Niveau(x) d'enseignement: 6 Numéro EDUQ : 9470484
Centre d'intérêt: 841

Lavoie, Claude.

Programmer et faire des mathématiques avec Mathématica / Claude Lavoie.
-- Montréal : Association des collèges privés du Québec, 1993.
233, [6] p. : ill. ; 28 cm. -- (Fonds A.C.P.Q. ; 30)

"Cette recherche a bénéficié du support financier du Fonds A.C.P.Q. 1991-92". Bibliogr.: p. [235]. Index.

Vedette(s)-secondaire(s): Association des collèges privés du Québec

Descripteur(s): Mathématiques * Logiciel * Programmation * Collégial (Ordre d'enseignement) * Universitaire (Ordre d'enseignement)

Identificateur(s) de sujet: Mathématica (Logiciel)

Présentation d'un logiciel (Mathématica) qui permet d'exécuter la partie technique des mathématiques avec un ordinateur. La première partie du document aborde les principales fonctions pré-construites dans le logiciel. La deuxième montre comment créer ses propres fonctions. La troisième porte sur la programmation proprement dite. La dernière partie utilise le logiciel pour résoudre certains problèmes de niveaux collégial et universitaire, plus particulièrement en Calcul I, II, et III.

Type de document:

FONDS
A.C.P.Q. 30

Claude LAVOIE

Professeur
Séminaire Saint-Augustin

Programmer et faire des mathématiques avec *Mathématica*

Juin 1993

Cette publication a bénéficié du support financier du Fonds A.C.P.Q. 1991-92

Copies de travail pour l'information des collègues membres de l'Association.
Tous droits de reproduction réservés par l'auteur.

Association
des collèges privés
du Québec

2540, boul. Pierre-Bertrand Est
Montréal (Québec) H2B 2B2

X07019270

94-70484

FONDS
A.C.P.Q. 30

Claude LAVOIE

Professeur
Séminaire Saint-Augustin

Programmer et faire des mathématiques avec *Mathématica*

Juin 1993

Cette recherche a bénéficié du support financier du Fonds A.C.P.Q. 1991-92

Copie de travail pour l'information des collèges membres de l'Association.
Tous droits de reproduction réservés par l'auteur.

Association
des collèges privés
du Québec

1940, boul. Henri-Bourassa Est
Montréal (Québec) H2B 1S2

AVERTISSEMENT

L'Association des collèges privés du Québec a créé le Fonds A.C.P.Q. pour encourager l'esprit de recherche dans l'enseignement et diffuser les résultats des recherches dans les collèges. Le support financier de l'Association ne couvre pas le coût réel des recherches et surtout ne paie pas à son vrai prix le travail intellectuel du chercheur. C'est pourquoi nous parlons seulement d'encouragement à la recherche. C'est pourquoi aussi les auteurs gardent tous les droits de publication et d'utilisation des résultats de leurs recherches, à moins que ces droits n'appartiennent, par un autre accord, au collège ou à un autre organisme pour lequel travaillent les auteurs.

Cependant, pour favoriser la diffusion de l'information sur l'état des recherches, les auteurs soutenus par le Fonds A.C.P.Q. s'engagent à mettre une copie de leurs travaux à la disposition de chaque collège membre de l'Association. Ainsi, l'Association n'est pas l'éditeur des rapports de recherches, qu'elle diffuse comme ils ont été présentés par les chercheurs. Pour toute autre utilisation des résultats des recherches, on doit d'abord obtenir l'autorisation des auteurs ou, selon le cas, de leur collège.

Montréal, juin 1993

Programmer et Faire des Mathématiques avec *Mathématica*

Claude Lavoie

Table des matières

Introduction 1

Première partie - Les calculs numériques

1.1 L'algèbre numérique 4

1.2 Les résultats exacts et approximatifs 5

1.3 Quelques fonctions mathématiques 6

1.4 L'algèbre symbolique

1.4.1 Les résultats antérieurs 10
1.4.2 Définir les variables 10
1.4.3 Les constantes 11

1.5 Les formules

1.5.1 Le calcul symbolique 12
1.5.2 Les étiquettes 13

1.6 Les opérateurs mathématiques dans *Mathematica*

1.6.1 Les opérateurs 14
1.6.2 Les dérivées 15
1.6.3 Les intégrales 16
1.6.4 Les sommes 17

1.7 Les équations

1.7.1 La différence entre = et == 18
1.7.2 Les limites 18
1.7.3 La résolution d'équations 19

1.8 Les méthodes numériques

1.8.1 Les méthodes approximatives 23

1.9 Les graphiques

1.9.1 Les graphiques 2D et 3D 24

Table des matières

Introduction 1

Première partie - Les calculs numériques

1.1 L'algèbre numérique 4

1.2 Les résultats exacts et approximatifs 5

1.3 Quelques fonctions mathématiques 6

1.4 L'algèbre symbolique

1.4.1 Les résultats antérieurs 10
1.4.2 Définir les variables 10
1.4.3 Les constantes 11

1.5 Les formules

1.5.1 Le calcul symbolique 12
1.5.2 Les étiquettes 13

1.6 Les opérateurs mathématiques dans *Mathematica*

1.6.1 Les opérateurs 14
1.6.2 Les dérivées 15
1.6.3 Les intégrales 16
1.6.4 Les sommes 17

1.7 Les équations

1.7.1 La différence entre = et == 18
1.7.2 Les limites 18
1.7.3 La résolution d'équations 19

1.8 Les méthodes numériques

1.8.1 Les méthodes approximatives 23

1.9 Les graphiques

1.9.1 Les graphiques 2D et 3D 24

Deuxième partie - Les procédures

2.1 Les procédures sans conditionnelles

2.1.1 Les procédures de base 31
2.1.2 Le langage 34
2.1.3 Quelques exemples de programmes 38

2.2 Les procédures avec conditionnelles

2.2.1 Les blocs de décision 47
2.2.2 Les procédures avec conditions 47

Troisième partie - La programmation

3.1 La programmation procédurale 53

3.2 La programmation fonctionnelle 68

3.3 La programmation mathématique 87

Quatrième partie - Les projets

Calcul I

Chapitre 1.

1.1 La définition de fonction 97

1.2 Le domaine de définition 99

1.3 La croissance d'une fonction 100

1.4 L'accroissement 102

1.5 Le maximum et le minimum 103

Chapitre 2.

2.1 La limite d'une fonction 105

2.2 Le nombre e 106

2.3 La limite à l'infini 107

2.4 La définition de limite 108

2.5 La définition de continuité 109

Chapitre 3.

3.1 La dérivée 118

3.2 La dérivée et la tangente 118

3.3 La dérivée et la vitesse 123

3.4 L'application de la dérivée 127

3.5 Les fonctions paramétriques 135

Chapitre 4.

4.1 Les différentielles 142

Calcul II

Introduction 146

Chapitre 1.

1.1 Les primitives et les intégrales 146

1.2 Les intégrales numériques 151

Chapitre 2.

2.1 Les séries 156

2.2 Les séries convergentes 157

Chapitre 3.

3.1 Les séries de MacLAURIN 160

Chapitre 4.

4.1 Les opérations avec les séries de puissances 164

Chapitre 5.

5.1 Le calcul de π 167

Calcul III

Introduction 171

Chapitre 1.

1.1 Les fonctions hyperboliques 171

1.2 Les intégrales numériques 175

Chapitre 2.

2.1 Les dérivées 178

2.2 Le gradient 180

2.3 Les fonctions contraintes 186

Chapitre 3.

3.1 Les intégrales multiples 196

3.2 Les volumes 198

Chapitre 4.

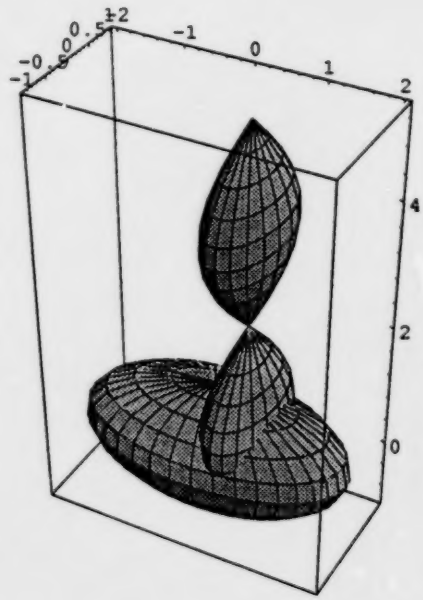
4.1 Les équations différentielles 212

4.2 Les équations différentielles linéaires d'ordre élevé 215

Chapitre 5.

5.1 Les séries de Fourier 223

Introduction



Mathematica est un logiciel qui exécute la partie technique des mathématiques avec un ordinateur. Il a été inventé pour les chercheurs, les ingénieurs, les architectes, les enseignants, les étudiants et toute autre personne qui utilisent les mathématiques dans leur travail. *Mathematica* est relativement facile à utiliser, tout se fait sous forme de fonctions (procédures). La principale difficulté est de traduire les symboles mathématiques dans le langage propre à *Mathematica* qui est sémantiquement près des mathématiques. Sauf quelques exceptions, les fonctions de *Mathematica* s'écrivent au long et commencent toujours par une majuscule. L'intégrale impropre de sin x / x sur l'intervalle 0 et π se traduit avec *Mathematica* de la façon suivante:

```
Limit[Integrate[Sin[x]/x, {x, epsilon, Pi}], epsilon -> 0]
```

Quand nous utilisons *Mathematica* comme calculateur, nous comptons surtout sur ses fonctions pré-construites. Mais il possède aussi un langage informatique avec lequel nous pouvons construire nos propres définitions, écrire nos programmes et travailler non seulement avec des nombres, mais aussi avec des expressions symboliques et des objets graphiques.

La première partie de ce livre présente les principales fonctions pré-construites dans *Mathematica*. Il y a plus de huit cents de ces fonctions dont on peut obtenir la description, les attributs et les options avec les touches ? et ?? Prenons par exemple, la fonction **Play[ ]**:

```
?Play

Play[f, {t, tmin, tmax}] joue un son dont l'amplitude est donnée
   par f considérée comme une fonction du temps t en secondes
   qui va de tmin à tmax.


??Play

Play[f, {t, tmin, tmax}] joue un son dont l'amplitude est donnée
   par f considérée comme une fonction du temps t en secondes
   qui va de tmin à tmax.

Attributes[Play] = {HoldAll, Protected}

Options[Play] =
  {Compiled -> True, Epilog -> {}, PlayRange -> Automatic,
   Prolog -> {}, SampleDepth -> 8, SampleRate -> 8192,
   DisplayFunction :> $SoundDisplayFunction}
```

Le point d'interrogation donne le format de la fonction, tandis que le double point d'interrogation donne en plus du format toutes les options et tous les attributs attachés à cette fonction. Il faut

consulter le livre de Stephen Wolfram: *Mathematica, A system for Doing Mathematics by Computer* pour avoir plus de détails.

La deuxième partie nous montre comment créer ses propres fonctions. Ce n'est qu'à la troisième partie que la programmation proprement dite débute. Tous les programmes ont été réalisés avec un Macintosh IIci. Ils vont du très facile au plus difficile. Ils n'ont pas une très grande utilité mathématique, ils sont surtout développés dans le but de montrer la puissance d'un langage récursif et de servir de tremplin pour produire des programmes plus spécifiques à nos besoins. Le livre de Roman Maeder: *Programming in Mathematica* décrit les méthodes pour programmer avec *Mathematica* en insistant sur le style qu'un programme doit avoir pour répondre à certaines normes afin de devenir partie intégrante du logiciel *Mathematica*.

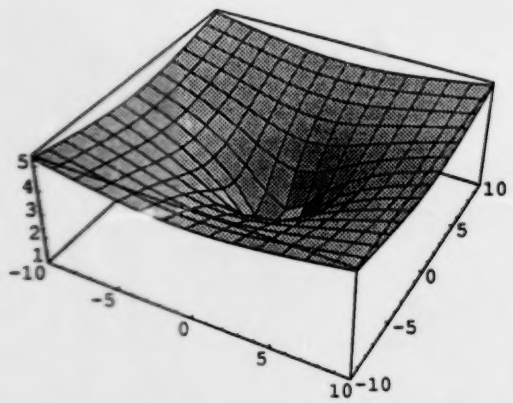
La dernière partie utilise *Mathematica* pour résoudre certains problèmes de niveaux collègial et universitaire, plus particulièrement en Calcul I, II et III.

Ce livre n'est pas un cours de mathématiques ou d'informatique, mais plutôt un schéma de travail pour établir un lien entre les méthodes traditionnelles de l'enseignement des mathématiques et les méthodes de l'avenir. Aujourd'hui, l'étudiant arrive en classe avec du matériel informatique aussi sophistiqué que ceux des bureaux météorologiques des années '70. Le marché produit et offre des portatifs de 4 Mo de RAM avec 40 Mo de ROM. Il est clair que demain nous travaillerons avec des étudiants qui auront à portée de doigts des logiciels titanesques. Et immanquablement, la méthodologie de l'enseignement des mathématiques devra changer. De nos jours, il y a des universités qui défendent les calculatrices perfectionnées et encore plus l'utilisation des ordinateurs. Mais ces restrictions ne dureront qu'un temps. Deux de nos étudiants l'an passé ont choisi telle université plutôt que telle autre parce qu'à l'université de leur choix ils pouvaient utiliser leur instrument informatique. Ils considéraient que les idées sont plus importantes que la technique. Vous pouvez dire que deux étudiants ce n'est pas beaucoup, oui, c'est vrai! mais ce sont des choses que nous n'entendions pas il y a quelques années.

Dans l'immédiat, nous devons donner à l'étudiant les moyens de se perfectionner aussi bien de façon traditionnelle qu'avec les méthodes à venir. L'important est qu'il puisse conceptualiser les situations selon ces deux méthodes. Il va de soi que nous préconisons toujours le constructivisme pour aborder les mathématiques. Et de plus, il nous semble que le logiciel *Mathematica* est l'instrument idéal pour mener à terme tous ces projets.

Première partie

Les calculs numériques



▫ 1.1 L'agèbre numérique

Nous pouvons faire de l'algèbre avec *Mathematica* de la même façon que nous faisons de l'arithmétique avec une calculatrice.

Pour trouver la somme de deux nombres on utilise tout simplement le + (*le point est le séparateur décimal*):

```
2.8 + 6.23
9.03
```

On se sert de / pour diviser et de ^ pour les puissances:

```
3.78 /7.89^4
0.000975402
```

Pour multiplier on se sert de * ou on laisse un espace.

```
3 * 74  23
5106
```

Les parenthèses permettent de grouper les expressions plus complexes:

```
(3+8)^4-2(5^7+10^3)
-143609
```

x y z ou x*y*z	**multiplier**
x+y+z	**additionner**
x^y	**élever x à la puissance y**
x - y	**soustraire**
x/y	**diviser**

Remarque.- Se rappeler que l'espace remplace la multiplication.

▫ 1.2 Les résultats exacts et approximatifs

Une calculatrice ordinaire nous donne une précision d'une dizaine de chiffres.
Mathematica nous offre la valeur exacte.

Nous obtenons la valeur exacte de 3^99, même si ce nombre contient 48 chiffres

```
3^99
```

```
171792506910670443678820376588540424234035840667
```

Avec la fonction abrégée //N nous pouvons obtenir une approximation décimale en notation scientifique.

```
3^99//N
```

$$1.71793\ 10^{47}$$

Mathematica exprime le résultat sous forme rationnelle simplifiée, quand c'est possible.

```
1/5 + 3/8
```

$$\frac{23}{40}$$

Si l'expression est suivi de //N on obtient une valeur numérique approximative de l'expression.

```
1/5 + 3/8 //N
```

```
0.575
```

Si on désire obtenir plus de décimales, il faut utiliser la fonction numérique sous sa forme complète: N[expression, nombre de chiffres].

> ***expression*** **//N donne une valeur**
>
> **numérique approximative de** ***expression***

▫ 1.3 Quelques fonctions mathématiques

Mathematica contient une très grande collection de fonctions mathématiques. Ce qui suit ne donne que les fonctions les plus usuelles.

La racine carrée:

```
Sqrt[1234]
```

```
Sqrt[1234]
```

L'exponentielle:

```
Exp[1234]
```

$$E^{1234}$$

Le logarithme naturel:

```
Log[1234]
```

```
Log[1234]
```

Le logarithme de base 5:

```
Log[5,1234]
```

$$\frac{\text{Log}[1234]}{\text{Log}[5]}$$

Les fonctions trigonométriques:

```
{Sin[Pi/2],Cos[Log[5.]],Tan[1.234],Sec[Pi/4]}
```

```
{1, -0.038632, 2.85603, Sqrt[2]}
```

Les fonctions trigonométriques réciproques:

```
{ArcSin[1],ArcCos[1/2],ArcTan[1],ArcSec[Sqrt[2.]]}
```

$$\left\{\frac{\text{Pi}}{2}, \frac{\text{Pi}}{3}, \frac{\text{Pi}}{4}, 0.785398\right\}$$

La fonction factorielle:

```
50!
```

```
30414093201713378043612608166064768844377641568960512000000000000
```

La fonction factorielle par sauts de deux:

```
50!!
```

```
52046984263666662269308108800000
```

La fonction valeur absolue:

```
Abs[-40]
```

```
40
```

La fonction de l'entier le plus près de x (*x réel*):

```
{Round[4.321],Round[4.567]}
```

```
{4, 5}
```

La fonction du plus grand entier inférieur ou égal à x (*x réel*):

```
{Floor[-4.321],Floor[+4.567]}
```

```
{-5, 4}
```

La fonction modulo:

```
{Mod[400,360],Mod[28,7],Mod[28,27]}
```

```
{40, 0, 1}
```

La fonction random (*donne un nombre au hasard entre 0 et 1, sauf indication contraire*):

```
{Random[],Random[Integer,{1,2}],Random[Real,{1,2}],
                                Random[Complex,{1+I,1-I}]}
```

```
{0.626906, 2, 1.54991, 1. + 0.798556 I}
```

La fonction maximum (*ou minimum*):

```
{Max[3,-2,1,6,-9],Min[3,-2,1,6,-9]}
```

```
{6, -9}
```

La fonction facteur premier:

```
FactorInteger[12345]
```

```
{{3, 1}, {5, 1}, {823, 1}}
```

Les arguments des fonctions sont encadrés par des parenthèses carrées.

La fonction factorielle par sauts de deux:

```
50!!
```

```
52046984263666662269308108800000
```

La fonction valeur absolue:

```
Abs[-40]
```

```
40
```

La fonction de l'entier le plus près de x (*x réel*):

```
{Round[4.321],Round[4.567]}
```

```
{4, 5}
```

La fonction du plus grand entier inférieur ou égal à x (*x réel*):

```
{Floor[-4.321],Floor[+4.567]}
```

```
{-5, 4}
```

La fonction modulo:

```
{Mod[400,360],Mod[28,7],Mod[28,27]}
```

```
{40, 0, 1}
```

La fonction random (*donne un nombre au hasard entre 0 et 1, sauf indication contraire*):

```
{Random[],Random[Integer,{1,2}],Random[Real,{1,2}],
                                Random[Complex,{1+I,1-I}]}
```

```
{0.626906, 2, 1.54991, 1. + 0.798556 I}
```

La fonction maximum (*ou minimum*):

```
{Max[3,-2,1,6,-9],Min[3,-2,1,6,-9]}
```

```
{6, -9}
```

La fonction facteur premier:

```
FactorInteger[12345]
```

```
{{3, 1}, {5, 1}, {823, 1}}
```

Les arguments des fonctions sont encadrés par des parenthèses carrées.

Les fonctions pré-construites dans *Mathematica*

commencent

toujours par une majuscule.

Les parenthèses ( ) sont utilisées pour regrouper les termes et non pas pour déterminer les arguments des fonctions.

Pour obtenir la valeur de $e^{3.56}$, on écrit

```
Exp[3.56]
35.1632
```

Ou bien, on écrit

```
E^3.56
35.1632
```

Remarque.- La lettre majuscule dans Exp[ ] et les parenthèses carrées qui entourent l'argument caractérisent les fonctions de *Mathematica*.

Constantes courantes dans *Mathematica* :

Pi	**3.14159...**
E	**2.7182818...**
Degree	**facteur de conversion des degrés en radians**
I	**la racine carrée de -1**
Infinity	**le symbole infini**

Voici quelques exemples qui donnent une bonne idée des constantes de *Mathematica*

```
N[Pi,50]
3.14159265358979323846264338327950288419716939937 51
N[E,50]
2.71828182845904523536028747135266249775724709 37
Sin[50 Degree]//N
0.766044
```

Nous pouvons nous servir de la fonction $f(x) = (1 + 1/x)^x$, de l'opérateur **Limit[ ]**, du symbole **Infinity** pour calculer la constante **E**.

```
Limit[(1+1/x)^x,x->Infinity]
E
```

Dans leur livre *Exploring Mathematics with Mathematica*, Jerry Glynn demandait à Theodore W. Gray: «Si vous étiez naufragé sur une île déserte et que vous ne puissiez prendre avec vous que huit fonctions, quelles seraient-elles? »

La réponse de Theodore fut: «En admettant que vous me donniez comme allant de soi des choses simples comme l'addition, les fonctions numériques comme Sin et les listes, voici les huit fonctions que je prendrais: **Table, Plot, Play, Do, Factor, Expand, Integrate et D**». Et il ajouta: «Il y a toutes sortes de choses que vous pouvez faire juste avec ces fonctions».

Dans les pages qui vont suivre ces fonctions seront largement utilisées, sauf la fonction **Play** qui demande beaucoup de temps d'exécution et de mémoire d'utilisation.

▫ 1.4 L'algèbre symbolique

• 1.4.1 Les résultats antérieurs

Lorsqu'un résultat est calculé par *Mathematica* il est placé dans la cellule numérotée **Out[***numéro***]**. On peut donc se servir de ce *numéro* pour faire de nouveaux calculs. Les contenus de ces cellules s'appellent les résultats antérieurs.

Voici les différentes façons de se servir des résultats antérieurs:

%	**le dernier résultat**
%%	**l'avant dernier résultat**
%n	**le résultat de la cellule Out[n]**

On fait calculer par *Mathematica* le nombre 89 à la puissance 3:

```
89^3
704969
```

Le dernier résultat est 704969. On n'est pas obligé de reproduire ce nombre à chaque fois, il suffit de taper %.

```
{%+5, Sin[%], Log[%]}//N
{704974., 0.948945, 13.4659}
```

L'opération précédente produit le même résultat que si on avait fait les calculs suivants:

```
{704969+5, Sin[704969], Log[704969]}//N
{704974., 0.948945, 13.4659}
```

• 1.4.2 Les variables

On appelle variable dans *Mathematica* toute expression pouvant remplacer une valeur, une fonction, une procédure, etc. Lorsqu'une valeur est attribuée à une variable elle l'est pour toute la durée de la session avec *Mathematica*, sauf si on vide la variable par la fonction **Clear[ ]** ou par **=.** ou si on l'écrase par une autre valeur ou expression.

• 1.4.3 Les constantes

Comment donner des valeurs aux variables:

x = constante	**x prend la valeur «constante»**
x = y = constante	**les deux symboles x et y prennent la valeur «constante»**
x =.	**x perd toute valeur**

Exemples:

En posant x = 8, x aura la valeur 8 tant que cette valeur ne sera pas changée:

```
x = 8
8
```

A chaque fois que nous utilisons x, *Mathematica* remplace x par 8:

```
x^4
4096
```

Nous pouvons donner à x une nouvelle valeur, i.e. écraser cette valeur par une autre:

```
x = 6 + 5
11
```

Le résultat précédent à la puissance 3:

```
%^3
1331
```

Nous pouvons aussi enlever à x toute valeur par x =. ou **Clear[x]**:

```
x=.
```

Après cette opération x est remis dans son état initial, i.e. *vide comme un contenant sans contenu:*

```
x
x
```

▫ 1.5 Les formules

• 1.5.1 Le calcul symbolique

Voici une des caractéristiques les plus importantes de *Mathematica* ; elle nous permet de faire des calculs symboliques (avec des variables) ainsi que des calculs numériques (avec des constantes). *Mathematica* façonne les formules algébriques tout aussi bien que les nombres.

Donnons à p la valeur 5

```
p = 5
```

```
5
```

Mathematica peut faire un calcul numérique en tenant compte de la valeur que nous venons de donner à p:

```
4 p - 7 p + 50
```

```
35
```

Soit q, une variable qui n'a reçu aucune valeur, *Mathematica* peut néanmoins simplifier 5q -3q +7:

```
5 q - 3 q + 7
```

```
7 + 2 q
```

Nous pouvons écrire n'importe quelle expression algébrique à l'aide des opérateurs acceptés par *Mathematica*. Il faut se rappeler que les espaces entre les nombres ou les variables remplacent l'opérateur «multiplication». Cependant, il faut faire attention de ne pas mettre un espace là où il n'en faut pas. L'exression **x y** signifie **x** multiplié par **y**, tandis que **xy** est une nouvelle variable ou un nouveau symbole qui n'a rien à voir avec le produit de x par y.

Exemples:

Mathematica réarrange et combine les termes selon les règles de l'algèbre:

```
x y + 2 x^2 y + y^2 x^2 - 2 y x
```

$$-(x\ y) + 2\ x^2\ y + x^2\ y^2$$

La fonction **Expand[ ]** développe une expression algébrique:

```
Expand[(x+y)^3]
```

$$x^3 + 3\ x^2\ y + 3\ x\ y^2 + y^3$$

La fonction **Factor[ ]** fait essentiellement l'inverse de **Expand[ ]**:

```
Factor[x^2 + 2 x y + y^2]
```

$$(x + y)^2$$

• 1.5.2 Les étiquettes

Il y a plusieurs façons d'utiliser les symboles. Jusqu'ici nous avons utilisé *Mathematica* avec des variables qui prenaient des valeurs numériques, mais nous pouvons aussi nous servir des variables pour représenter des unités physiques. Lorsque les variables sont vides, i.e. sans valeur particulière, *Mathematica* les traite comme des expressions symboliques. Par exemple, quand nous divisons m par s, si m et s n'ont aucune valeur initiale, *Mathematica* écrira:

```
Clear[m, s];
m/s
```

$$\frac{m}{s}$$

Nous allons nous servir de cette caractéristique spéciale à *Mathematica* pour coller une étiquette aux variables algébriques.

D'abord il faut vider m de toute valeur par:

```
m =.
```

Le symbole **m** joue le rôle d'étiquette et supposons qu'il remplace «mètre»:

```
39 m
```

$$39\ m$$

On peut de cette façon additionner les longueurs (% remplace 39 m):

```
% + 7.8 m
```

$$46.8\ m$$

ou trouver la vitesse (% remplace 46.8 m):

```
%/(22 s)
```

$$\frac{2.12727\ m}{s}$$

ou faire la conversion «mètres» en «pieds» (% remplace 2.12727 m/s):

```
%/.m->3.28084 pieds
```

$$\frac{6.97924\ pieds}{s}$$

▫ 1.6 Les opérateurs mathématiques dans *Mathematica*

• 1.6.1 Les opérateurs

Si une fonction est continue et dérivable, *Mathematica* possède les règles de transformation pour calculer la dérivée de ce type de fonctions.

Avec *Mathematica* on peut dériver une expression symbolique et obtenir une formule comme résultat.

Pour dériver x à la puissance n, on écrit:

```
D[x^n, x]
```

$$n\ x^{-1+n}$$

Ou avec un exemple légèrement plus compliqué:

```
D[x^3 Log[x^2+b], x]
```

$$\frac{2\ x^4}{b + x^2} + 3\ x^2\ \text{Log}[b + x^2]$$

Ou bien:

```
D[Sin[Cos[x]]^Log[Tan[x]], x]
```

$$-(\text{Cos}[\text{Cos}[x]]\ \text{Log}[\text{Tan}[x]]\ \text{Sin}[x]\ \text{Sin}[\text{Cos}[x]]^{-1+\text{Log}[\text{Tan}[x]]}) +$$
$$\text{Csc}[x]\ \text{Log}[\text{Sin}[\text{Cos}[x]]]\ \text{Sec}[x]\ \text{Sin}[\text{Cos}[x]]^{\text{Log}[\text{Tan}[x]]}$$

Les principaux opérateurs sont:

D[f, x]	**la dérivée de f**
Integrate[f, x]	**la primitive de f**
Integrate[f, {x, xmin, xmax}]	**l'intégrale de xmin à xmax**

Sum[f, {i, imin, imax}]	**la sommation de f passant par toutes les itérations de imin à imax**
Product[f, {imin, imax}]	**le produit de f passant par toutes les itérations de imin à imax**
Limit[f, x->x_0]	**la limite de f quand x tend vers x_0**

• 1.6.2 Les dérivées

La fonction **D[x^n, x]** exécute une dérivée partielle, où n est constant relativement à x.

Mathematica possède une autre fonction, appelée **Dt[ ]**, laquelle calcule la dérivée totale. On se sert de cette fonction, si les variables dépendent d'une variable commune.

Dt[ ] donne la dérivée totale, ce qui laisse supposer que n et x ne sont pas indépendants (dans l'exemple qui suit t joue le rôle d'intermédiaire).

```
Dt[x^n,t]
```

$$n\,x^{-1+n}\,\mathrm{Dt}[x,\,t] + x^{n}\,\mathrm{Dt}[n,\,t]\,\mathrm{Log}[x]$$

D[f, x]	**la dérivée partielle de f**
D[f, x1, x2, ...]	**les dérivées multiples**
D[f, {x, n}]	**la dérivée d'ordre n**
Dt[f]	**la différentielle totale df**
Dt[f, x]	**la dérivée totale**

Remarque.- Soit **f[x]** une fonction, **f'[x]** représente la dérivée première et **f''[x]** la dérivée seconde.

Exemples:

```
f[x_]:= Cos[x^2 + 3 x + 5]^7
```

```
f'[x]
```

$$-7\ (3 + 2\ x)\ \text{Cos}[5 + 3\ x + x^2]^6\ \text{Sin}[5 + 3\ x + x^2]$$

```
f''[x]
```

$$-7\ (3 + 2\ x)^2\ \text{Cos}[5 + 3\ x + x^2]^7 -$$
$$14\ \text{Cos}[5 + 3\ x + x^2]^6\ \text{Sin}[5 + 3\ x + x^2] +$$
$$42\ (3 + 2\ x)^2\ \text{Cos}[5 + 3\ x + x^2]^5\ \text{Sin}[5 + 3\ x + x^2]^2$$

• 1.6.3 Les intégrales

Pour trouver la primitive d'une fonction, on écrit:

```
Integrate[x^n,x]
```

$$\frac{x^{1 + n}}{1 + n}$$

Pour calculer l'intégrale d'une fonction entre les bornes 2 et 5, on ajoute les bornes dans une liste:

```
Integrate[x^3,{x,2,5}]
```

$$\frac{609}{4}$$

Pour calculer l'intégrale d'une fonction qui n'a pas de primitive entre 0 et π /2, on se sert de **NIntegrate[ ]**, l'intégrale numérique:

```
NIntegrate[Sin[x]/x,{x,0,Pi/2}]
```

```
1.37076
```

Pour calculer une intégrale double, on répète l'opération intégrale:

```
Integrate[Integrate[x^2 y^3,x],y]
```

$$\frac{x^3\ y^4}{12}$$

Ou on précise les variables, quoique ce procédé ne fonctionne pas toujours avec les intégrales triples:

```
Integrate[x^2 y^3,x,y]
```

$$\frac{x^3\ y^4}{12}$$

Pour calculer une intégrale triple, on reproduit l'opérateur intégrale trois fois:

```
Integrate[Integrate[Integrate[Cosh[x],
                              {x,0,y+z}],{y,0,2 z}],{z,1,2}]
```

$$\text{Sinh}[1] - \text{Sinh}[2] - \frac{\text{Sinh}[3]}{3} + \frac{\text{Sinh}[6]}{3}$$

• 1.6.4 Les sommes

La somme de i = 1 à i = 7 de x^i / i! se fait ainsi:

```
Sum[x^i/i!,{i,1,7}]
```

$$x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} + \frac{x^6}{720} + \frac{x^7}{5040}$$

Si la limite inférieure est égale à un, on peut l'omettre:

```
Sum[x^i/i!,{i,7}]
```

$$x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} + \frac{x^6}{720} + \frac{x^7}{5040}$$

Si on veut augmenter l'indice par sauts de deux, on écrit 2 à la fin de la liste des indices:

```
Sum[x^i/i!,{i,1,7,2}]
```

$$x + \frac{x^3}{6} + \frac{x^5}{120} + \frac{x^7}{5040}$$

Remarque.- Si on part avec 0, au lieu de 1, on aura des exposants pairs au lieu d'impairs.

Les produits se composent de la même façon que les sommes:

```
Product[x + i, {i, 1, 4}]
(1 + x) (2 + x) (3 + x) (4 + x)
```

Voici un exemple peu familier qui donne un nombre astronomique:

```
Product[(i^i + 1)^i,{i,1,5}]
142929363634936111921593716213760 0
```

Pour composer une somme de sommes, on écrit:

```
Sum[i j,{i,0,7},{j,1,3}]
168
```

▫ 1.7 Les équations

• 1.7.1 La différence entre = et ==

Les deux symboles = et == se ressemblent. Il ne faut pas les confondre, car l'un assigne une expression et l'autre vérifie une expression:

x = y force x à prendre la valeur y (= *caractérise un remplacment*)

x == y vérifie si x est bien égal à y (== *est une équation*).

Pour tester si **2 + 3** et **5** sont égaux on utilise ==.
Le résultat est le symbole **True**, si == est affirmatif.

```
2 + 3 == 5
```

```
True
```

Il est essentiel de ne pas confondre x = y avec x == y. Le symbole = impose la valeur y à la variable x, tandis que le symbole == teste si les deux membres d'une même équation sont égaux.

• 1.7.2 Les limites

Soit l'expression (1 - cos x) / x:

```
z = (1 - Cos[x])/x
```

$$\frac{1 - \text{Cos}[x]}{x}$$

Si nous remplaçons x par 0, l'expression devient 0/0 et nous obtenons une indétermination précédée d'un message écrit par *Mathematica*:

```
z/.x->0
```

```
                                    1
Power::infy: Infinite expression - encountered.
                                    0

Infinity::indet:
   Indeterminate expression 0 ComplexInfinity encountered.

Indeterminate
```

La valeur numérique de (1 - cos x) / x pour des valeurs de x près de 0 donne un résultat près de 0, si x est petit et non nul.

```
z/. x->0.00001
```

$$5.\ 10^{-6}$$

Pour calculer la limite de (1 - cos x) / x quand x s'approche de 0, on écrit:

```
Limit[z, x->0]
0
```

En bref:

Limit[f[x], x->x_0]	la limite de f[x] quand x tend vers x_0

• 1.7.3 La résolution d'équations

Une expression de la forme x^2 - 3 x - 9 == 0 représente une équation dans *Mathematica* . Il arrive souvent que nous ayons besoin de résoudre ce genre d'équation, afin de trouver les valeurs de x pour lesquelles l'équation est vérifiée.

Pour résoudre l'équation quadratique x^2 - 3x - 9 == 0 on utilise **Solve[ ]**. Les solutions sont données sous forme de remplacements.

```
Solve[x^2 -3x -9 == 0,x]
```

$$\{\{x \to \frac{3 + 3\ \text{Sqrt}[5]}{2}\},\ \{x \to \frac{3 - 3\ \text{Sqrt}[5]}{2}\}\}$$

Les valeurs numériques des solutions sont obtenues par la fonction **N[,10]**, où 10 représente le nombre de chiffres significatifs.

```
N[%,10]
{{x -> 4.854101966}, {x -> -1.854101966}}
```

Nous pouvons prendre la liste des remplacements produite par **Solve[ ]** et utiliser /. pour les substituer dans une expression faisant intervenir x.

```
x^2-3x-9/.%
```

$$\left\{-9 - \frac{3\ (3 + 3\ \text{Sqrt}[5])}{2} + \frac{(3 + 3\ \text{Sqrt}[5])^2}{4},\right.$$
$$\left. -9 - \frac{3\ (3 - 3\ \text{Sqrt}[5])}{2} + \frac{(3 - 3\ \text{Sqrt}[5])^2}{4}\right\}$$

Et si on désire faire une vérification sachant que la simplification de l'expression précédente doit donner {0, 0}:

```
Simplify[%]
{0, 0}
```

Solve[ ] essaie toujours de solutionner les équations sous forme de formules. Cependant, si pour des équations suffisamment complexes une formule ne peut pas être trouvée, *Mathematica* la remplace par la fonction **Roots[ ]** de laquelle nous pouvons obtenir des racines approximatives.

Si nous avons une équation à une variable et si la plus haute puissance est au plus quatre, alors *Mathematica* peut nous donner la solution sous forme de formules. Mais si la plus haute puissance est cinq ou plus, on sait qu'il est algébriquement impossible de trouver des formules pour toutes les solutions, sauf quelques cas particuliers.

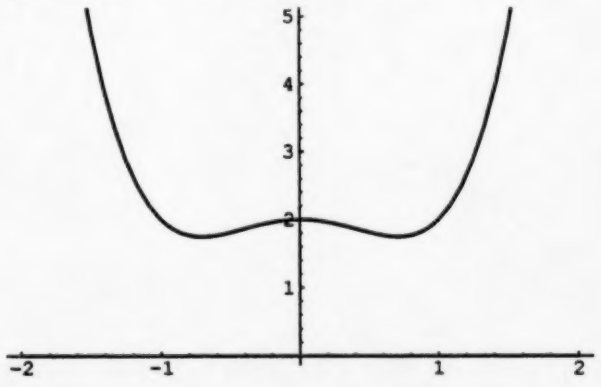
Mathematica peut toujours résoudre les équations algébriques d'une variable quand la plus haute puissance est plus petite que cinq, même si les racines sont complexes.

```
Solve[x^4 - x^2 + 2 ==0, x]
```

$$\left\{\left\{x \to \frac{\text{Sqrt}[1 + I\ \text{Sqrt}[7]]}{\text{Sqrt}[2]}\right\}, \left\{x \to -\left(\frac{\text{Sqrt}[1 + I\ \text{Sqrt}[7]]}{\text{Sqrt}[2]}\right)\right\},\right.$$
$$\left.\left\{x \to \frac{\text{Sqrt}[1 - I\ \text{Sqrt}[7]]}{\text{Sqrt}[2]}\right\}, \left\{x \to -\left(\frac{\text{Sqrt}[1 - I\ \text{Sqrt}[7]]}{\text{Sqrt}[2]}\right)\right\}\right\}$$

Si on trace le graphe de la fonction quadratique f[x]=x^4 -x^2 + 2, on voit qu'il ne coupe pas l'axe des x, alors il ne peut pas y avoir de racines réelles.

```
Plot[{0,x^4 -x^2 + 2} ,{x,-2,2}]
```



Dans le graphique ci-dessus la fonction n'est définie que pour des valeurs de l'intervalle [-2, 2]. C'est cependant suffisant pour se rendre compte que le graphe ne coupera pas l'axe des x, même si l'ensemble de définition de f est l'ensemble des nombres réels.

Il y a des équations où il est algébriquement impossible de trouver des formules explicites donnant les solutions. *Mathematica* utilise la fonction **Roots[ ]** pour représenter la solution approximative.

```
Solve[x^5 - x + 9 == 0,x]
```

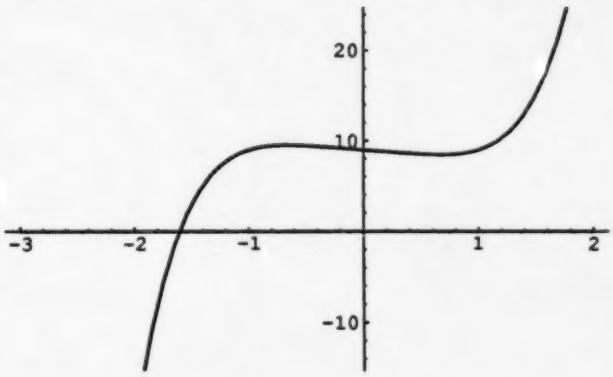
```
{ToRules[Roots[-x + x^5 == -9, x]]}
```

Nous ne pouvons pas trouver de formules explicites, nous avons quand même accès aux solutions numériques approximatives à 10 chiffres significatifs.

```
N[%,10]
```

```
{{x -> -1.603579514}, {x -> -0.4377638189 - 1.50837279 I},
  {x -> -0.4377638189 + 1.50837279 I},
  {x -> 1.239553576 - 0.8594622641 I},
  {x -> 1.239553576 + 0.8594622641 I}}
```

```
Plot[{x^5 - x + 9, 0},{x,-3,2}]
```



Le graphique nous montre bien que la racine réelle se trouve à l'intersection du graphe et de l'axe des x.

La fonction **Solve[ ]** n'est pas limitée aux simples équations, elle peut également résoudre des systèmes d'équations. Il suffit de lui donner la *liste* des variables ainsi que la *liste* des équations dont nous cherchons la solution. Voici une *liste* de deux équations qui doivent être résolues en tenant compte de la *liste* des variables x et y (*on appelle liste tout ce qui se trouve entre les deux accolades { et }*).

```
Solve[{a x + y == 0, 2x + (1-a)y == 1}, {x, y}]
```

$$\{\{x \to -\left(\frac{1}{-2 + a - a^2}\right),\ y \to \frac{a}{-2 + a - a^2}\}\}$$

Pour éliminer y entre les deux équations, nous pouvons nous servir de **Eliminate [ ]**. On retrouve ainsi une équation sans y.

```
Eliminate[{a x + y == 0, 2x + (1- a)y == 1},y]
```

$$-(a\ x) + a^2\ x == 1 - 2\ x$$

Remarque. - Pour indiquer que l'ensemble des solutions est vide, *Mathematica* écrit { }. Cependant, il peut écrire { } pour une toute autre raison, il peut arriver qu'en utilisant des fonctions réciproques il ne soit pas en mesure de donner les solutions. Dans ce cas il écrira { } quand même.

▫ 1.8 Les méthodes numériques

On doit avoir recours aux méthodes numériques quand il n'est pas possible de trouver des résultats symboliques exacts.

Il semble que *Mathematica* soit muni des méthodes de calcul les plus récentes afin d'effectuer les opérations le plus rapidement possible.

Parmi ces méthodes nous retrouvons:

- l'approximation numérique d'une intégrale, d'une somme et des racines d'un polynôme;
- la recherche d'une solution numérique dont le départ est x_0;
- la recherche d'une valeur minimale d'une fonction dont le départ est x_0.

• 1.8.1 Les méthodes approximatives

N[m, 20]	la valeur numérique de l'expression m à 20 décimales
NIntegrate[f, {x, xmin, xmax}]	approximation numérique d'une intégrale
NSum[f, {i, imin, Infinity}]	approximation numérique de Σ f entre i=imin et l'infini
NRoots[polynôme == 0, x]	approximation numérique des racines d'un polynôme
FindRoot[mg == md, {x, x_0}]	recherche d'une solution numérique dont le départ est x_0
FindMinimum[f, {s, x_0}]	cherche la valeur minimale de f dont le départ est x_0

Remarque.- Les expressions mg et md dans **FindRoot**[mg == md, {x, x_0}] désignent respectivement le membre de gauche et le membre de droite de l'équation.

▫ 1.9 Les graphiques

• 1.9.1 Les graphiques 2D et 3D

Mathematica peut tracer des graphiques en 2 D ou en 3 D.

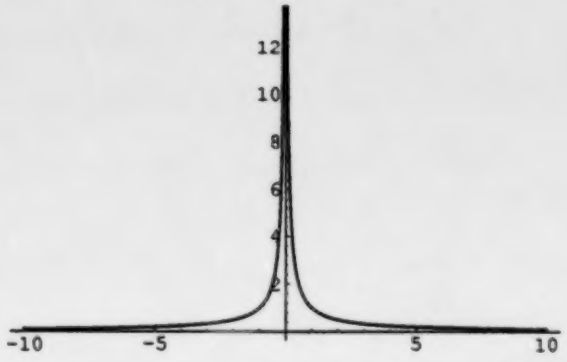
Le tracé de base:

Plot[f, {x, xmin, xmax}]	trace le graphe de f de xmin à xmax;
Plot[{f, g, ...}, {x, xmin, xmax}]	trace plusieurs fonctions superposées sur le même graphique;
Plot3D[f[x,y], {x, xmin, xmax},{y, ymin, ymax}]	trace f en trois dimensions.

Voici une suite d'exemples qui illustrent la fonction **Plot[ ]**

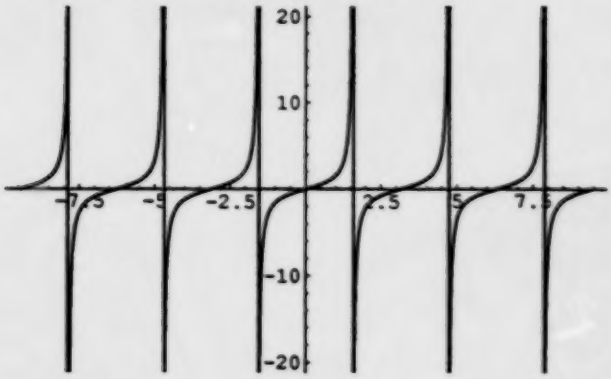
Pour tracer le graphe de y = 1 / |x| entre -10 et 10;

```
Plot[1/Abs[x], {x, -10, 10}]
```



Naturellement, on reçoit un message d'erreur, puisque la fonction logarithmique n'est pas définie à x = 0. Avec les fonctions qui ont des points de discontinuité *Mathematica* produit quand même le graphique tout en essayant de choisir une échelle acceptable.

```
Plot[Tan[x], {x,-3Pi,3Pi}]
```



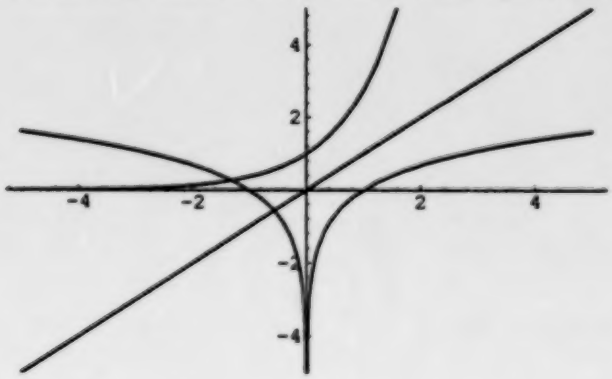

```
-Graphics-
```

Il est aussi possible de superposer une *liste* de fonctions. Par exemple, si on veut mettre sur un même graphique f[x] = Exp[x], g[x] = x et h[x] = Log[|x|], il suffit de placer toutes les fonctions dans une *liste* et de procéder de façon ordinaire.

```
?PlotRange
```

```
PlotRange est une option graphique qui spécifie quels sont les
points qu'on doit inclure  dans un tracé de fonction.
```

```
Plot[{Exp[x],x,Log[Abs[x]]},{x,-5,5},PlotRange->{-5,5}]
```

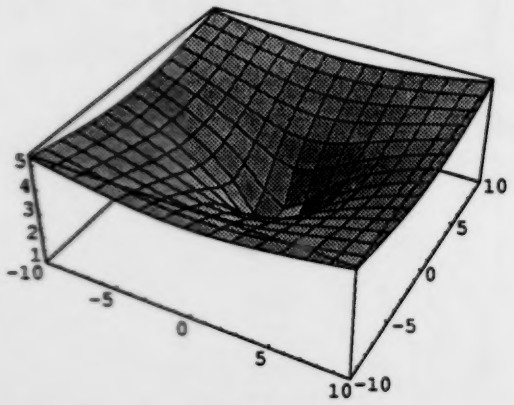



```
-Graphics-
```

On s'est servi de l'option **PlotRange[ ]** pour balancer les graphes afin d'obtenir un tracé plus symétrique.

Pour tracer le graphique d'une fonction de deux variables on se sert de la fonction **Plot3D[ ]**.

```
Plot3D[Log[x^2+y^2],{x,-10,10},{y,-10,10}]
```

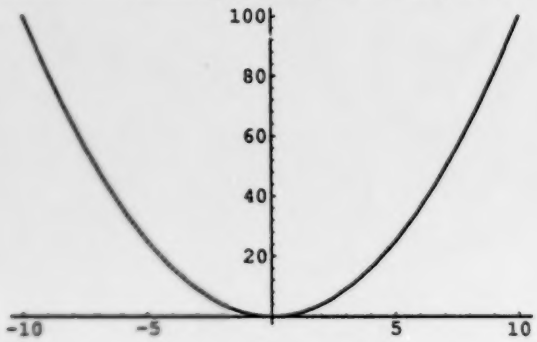



```
-SurfaceGraphics-
```

Nous pouvons aussi tracer des fonctions paramétriques:

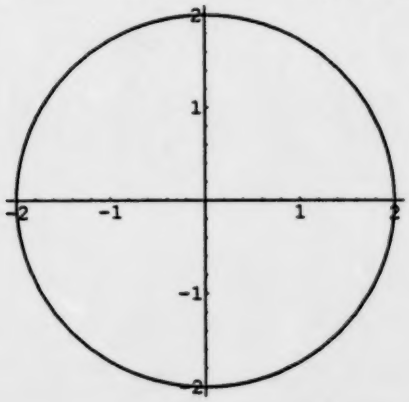
Une parabole:

```
ParametricPlot[{t,t^2},{t,-10,10}]
```



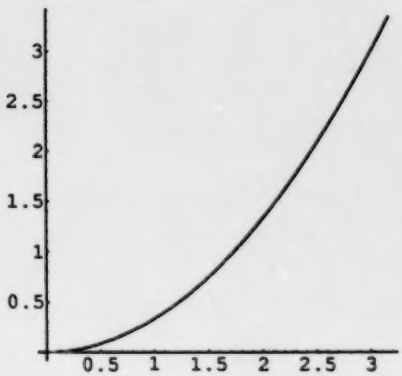
Un cercle:

```
ParametricPlot[{2 Cos[t],2 Sin[t]},{t,-Pi,Pi},AspectRatio -> 1]
```



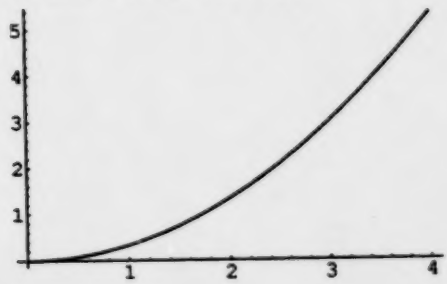
La fonction y = x^(2 / 3) définie paramétriquement par:

```
ParametricPlot[{Sqrt[t],t^1/3},{t,0,10},AspectRatio -> 1]
```



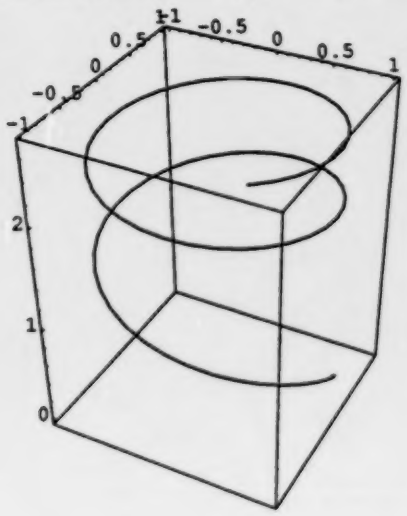
On peut vérifier avec l'équation cartésienne que les deux graphiques sont les mêmes.

```
Plot[x^2/3,{x,0,4}]
```



Afin de bien voir les fonctions de ligne en 3 D *Mathematica* les place dans un parallélépipède. La fonction suivante s'enroule sur un cylindre, mais le pas est logarithmique.

```
ParametricPlot3D[{Sin[t],Cos[t],Log[t]},{t,1,15}]
```



Voici un enroulement cylindrique d'une bande où le graphique est dessiné sans le parallélépipède et sans les axes:

```
ParametricPlot3D[{2t - u, Sin[u],Cos[u]},{u,0,2Pi},{t,-2,2},
Boxed -> False,Axes->False,LightSources ->
    {{{1., 0., 1.}, RGBColor[.8, 0, 0]},
     {{1., 1., 1.}, RGBColor[0, 1, 0]},
     {{0., 1., 1.}, RGBColor[0, 0, .8]}}]
```

Deuxième partie

Les procédures

□ 2.1 Les procédures sans conditionnelles

• 2.1.1 Les procédures de base

Quand nous échafaudons des calculs dans *Mathematica* , il peut arriver que nous ayons besoin d'aligner une suite de fonctions pour obtenir le résultat final. Une procédure simple est une suite d'opérations placées les unes à la suite des autres et séparées par des points-virgules. Les fonctions dans une procédure sont évaluées à la queue leu leu, le résultat final étant la valeur de la dernière expression calculée.

Chaque élément de la procédure est calculé dans l'ordre descendant. La dernière expression donne le résultat final:

```
s = {x,y,z};
t = s[[1]]^2 + s[[2]]^2 + s[[3]]^2;
v = Sqrt[t];
w= s/v;
ArcCos[w]
```

$$\{\text{ArcCos}[\frac{x}{\text{Sqrt}[x^2 + y^2 + z^2]}], \text{ArcCos}[\frac{y}{\text{Sqrt}[x^2 + y^2 + z^2]}], \text{ArcCos}[\frac{z}{\text{Sqrt}[x^2 + y^2 + z^2]}]\}$$

Nous pouvons d'une procédure faire une fonction. Cette fonction transforme les composantes d'un vecteur (x, y, z) en cosinus directeurs donnés en radians.

```
f[x_,y_,z_]:=(s = {x,y,z};
t = s[[1]]^2 + s[[2]]^2 + s[[3]]^2;
v = Sqrt[t];
w= s/v;
ArcCos[w])
```

```
f[1,2,3]
```

$$\{\text{ArcCos}[\frac{1}{\text{Sqrt}[14]}], \text{ArcCos}[\text{Sqrt}[\frac{2}{7}]], \text{ArcCos}[\frac{3}{\text{Sqrt}[14]}]\}$$

L'image de f[1,2,3] par cette nouvelle fonction donne les cosinus directeurs du vecteur (1, 2, 3).

Avec la fonction que nous venons de définir, nous pouvons composer une autre fonction qui convertit les cosinus directeurs en degrés en multipliant f(x, y, z) par 180/π.

```
g[x_,y_,z_]:= f[x,y,z] 180/Pi //N
```

Voici les angles que fait le vecteur (1, 2, 3) avec la partie positive des axes en 3D:

```
g[1,2,3]
{74.4986, 57.6885, 36.6992}
```

Même genre de calcul avec le vecteur (3, - 5, 4):

```
g[3,-5,4]
{64.8959, 135., 55.5501}
```

Les procédures ne sont pas uniques. Voici une autre façon d'écrire la même fonction que la fonction g, mais avec une écriture complètement différente. Elle nous donne les cosinus directeurs du vecteur (1, 2, 3) en degrés avec une précision de 15 chiffres significatifs.

```
sf[x_,y_,z_]:=N[ArcCos[{x,y,z}/({x,y,z}/.{x,y,z}->
                                Sqrt[x^2+y^2+z^2])]
                                180/Pi,15]

sf[1,2,3]
{74.498640433063, 57.6884667625762, 36.6992252004899}
```

Dans un autre ordre d'idées, voici une procédure, appelée **dms[ ]**, qui convertit les degrés avec fraction décimale en degrés, minutes et secondes.

```
dms[x_]:=(int=(x-Floor[x])60;
          degre=Floor[x];
          minute=Floor[int];
          seconde=(int-Floor[int]) 60;
          Print[degre, "° ", minute, "' ", seconde, " '' "])
```

On vérifie que la conversion de 45.477° en degrés, minutes et secondes est 45 degrés, 28 minutes et 37.2 secondes:

```
dms[45.477]
45° 28' 37.2 ''
```

Quand les procédures sont acceptées par *Mathematica*, il est toujours possible de les composer (*dans le sens fonctionnel*) pour obtenir d'autres fonctions, d'autres procédures.

Pour faire suite à ce qui précède, voici une fonction, appelée **trans[a,b,c]**, qui est la composée de **dms[ ]** et **sf[ ]**:

```
trans[a_,b_,c_]:=Do[ dms[ sf[a,b,c][[i]] ],{i,1,3}]
```

Donc, à partir d'un vecteur (x, y, z), la fonction **trans[ ]** nous donne les cosinus directeurs en degrés, minutes et secondes.

```
trans[3,-5,4]
```

```
64° 53' 45.2751 ''
                  -13
135° 0' 2.9976 10     ''
55° 33' 0.352843 ''
```

Avec le vecteur (1, 1, 1), (*celui qui se situe au sommet du cube placé dans le premier octant*), on trouve:

```
trans[1,1,1]
```

```
54° 44' 8.19714 ''
54° 44' 8.19714 ''
54° 44' 8.19714 ''
```

Quand on utilise *Mathematica* au niveau des procédures, il peut arriver qu'on ait besoin de variables temporaires, ordinairement appelées variables locales.

On peut utiliser la construction **Block[{x, y, ...}, procédure]** afin de spécifier la *liste* des variables locales dans une procédure particulière. Dans le **Block** qui suit, {x, y, z} représente la *liste* des variables locales. Le **Block** est composé à partir des fonctions **sf** et **dms** pour donner la fonction **trans**. Bref, le ***Block procédure*** nous donne à partir d'un vecteur (a, b, c) les cosinus directeurs en degrés, minutes et secondes. C'est la méthode standard pour écrire une procédure dans *Mathematica*.

```
Clear[a,b,c,x,y,z,i,dms,sf,trans]
```

Clear[ ] efface toutes les variables utilisées dans les procédures précédentes afin de composer une nouvelle fonction **trans[ ]**.

```
trans[a_,b_,c_]:=
     Block[{x,y,z,i},
          sf[x_,y_,z_]:=N[ArcCos[{x,y,z}/({x,y,z}/.{x,y,z}->
               Sqrt[x^2+y^2+z^2])] 180/Pi,15];
          dms[x_]:=(int=(x-Floor[x])60;
               degre=Floor[x];
               minute=Floor[int];
               seconde=(int-Floor[int]) 60;
               Print[degre, "° ", minute, "' ", seconde, " '' "]);
          Do[dms[sf[a,b,c][[i]]],{i,1,3}]
     ]
```

On vérifie la nouvelle fonction **trans[ ]** avec un des calculs fait plus-haut:

```
trans[1,1,1]

54° 44' 8.19714 ''
54° 44' 8.19714 ''
54° 44' 8.19714 ''
```

De plus:

```
dms[47.718]

47° 43' 4.8 ''
```

Non seulement la fonction **trans[ ]** exécute la tâche pour laquelle elle a été créée, mais les fonctions **sf[ ]** et **dms[ ]** sont également réactivées.

• 2.1.2 Le langage

Mathematica est aussi un langage de programmation. On y trouve toutes les fonctions pour faire de la programmation structurée, entre autres les fonctions qui permettent de faire des *boucles*.

Certaines opérations que l'on peut utiliser avec les *boucles:*

i++	augmente la valeur de i de 1
i--	diminue la valeur de i de 1
++i	pré-augmente la valeur de i
--i	pré-diminue la valeur de i
i += di	additionne di à la valeur de i
i -= di	soustrait di à la valeur de i

x *= c	multiplie x par c
x /=c	divise x par c
{x, y} = {y, x}	substitue les valeurs de x et de y
Prepend[*liste*, s]	ajoute s au début de *liste*
Append[*liste*, s]	ajoute s à la fin de *liste*

PrependTo[*liste*, s]	est équivalent à	s = **Prepend**[*liste*, s]
AppendTo[*liste*, s]	est équivalent à	s = **Append**[*liste*, s]

Voici quelques définitions et exemples:

Do exécute *expression* avec i variant de 1 à imax par sauts de un:

Do[*expression*, {i, imax}]

```
Do[Print[i,"  sin[", i,"] = ",Sin[i]//N], {i,5}]

1  sin[1] = 0.841471
2  sin[2] = 0.909297
3  sin[3] = 0.14112
4  sin[4] = -0.756802
5  sin[5] = -0.958924
```

Do exécute *expression* avec i variant de imin à imax par sauts de ipas:

Do[*expression*, {i, imin, imax, ipas}]

```
Do[Print["sin[", i"] = ",Sin[i]//N], {i,2,10,2}]

sin[2 ] = 0.909297
sin[4 ] = -0.756802
sin[6 ] = -0.279415
sin[8 ] = 0.989358
sin[10 ] = -0.544021
```

Do exécute *expression n* fois:

Do[*expression*, {n}]

```
Do[Print[Mathematica ], {5}]

Mathematica
Mathematica
Mathematica
Mathematica
Mathematica
```

Nest applique **f** à *expression n* fois:

Nest[f, *expression*, **n]**

```
Nest[g, a, 5]

g[g[g[g[g[a]]]]]

Nest[f, Log[a], 3]

f[f[f[Log[a]]]]
```

FixedPoint part avec *expression*, et applique **f** jusqu'à ce que le résultat ne change plus:

FixedPoint[f, *expression*]

```
FixedPoint[Function[x, (1-x)/x], 2.0]

-1.61803
```

FixedPointList[ ] agit comme **FixedPoint[ ]** en donnant en plus la liste des étapes qui montre que le résultat ne change plus:

```
FixedPointList[Function[x, (1-x)/x], 2.0]

{2., -0.5, -3., -1.33333, -1.75, -1.57143, -1.63636, -1.61111,
  -1.62069, -1.61702, -1.61842, -1.61789, -1.61809, -1.61801,
  -1.61804, -1.61803, -1.61804, -1.61803, -1.61803, -1.61803,
  -1.61803, -1.61803, -1.61803, -1.61803, -1.61803, -1.61803,
  -1.61803, -1.61803, -1.61803, -1.61803, -1.61803, -1.61803,
  -1.61803, -1.61803, -1.61803, -1.61803, -1.61803, -1.61803,
  -1.61803, -1.61803, -1.61803, -1.61803, -1.61803}
```

While évalue expression jusqu'à ce que *test* soit vrai:

While[*test*, expression]

```
m=0;i=0;
While[i<5,Print[i," ",m//N];m = Exp[i+1];i++];
m//N

0 0
1 2.71828
2 7.38906
3 20.0855
4 54.5982

148.413
```

La fonction **While[ ]** dans la procédure précédente fait imprimer les valeurs de i et m. Au départ i et m sont tous les deux égaux à 0. Le dernier calcul se fait quand i = 4 et m = 54.5982. À ce moment la fonction **While[ ]** a fini sont travail en ce qui concerne m et augmente i de 1. C'est pour cette raison que lorsque la procédure demande m à la dernière ligne la valeur 148.413 est donnée. On peut en faire la vérification avec le calcul suivant:

```
Exp[5]//N

148.413
```

For évalue *départ*, et alors évalue respectivement *pas* et *test* jusqu'à ce que *test* soit faux:

For[*départ*, *test*, *pas*, expression]

```
m=0;
For[i=0,i<5,i++,Print[i," ",m//N];m = Exp[i+1]];
m//N

0 0
1 2.71828
2 7.38906
3 20.0855
4 54.5982

148.413
```

Rien ne nous dit quelles fonctions utiliser; **Do**, **For**, **While**..., c'est le génie créateur de l'analyste qui doit décider du choix des fonctions et de l'ordre du déroulement. L'étude et la pratique sont les clés pour apprendre à maîtriser les procédures.

On peut analyser le petit problème suivant:

Faire une *boucle* qui va de deux jusqu'à cinq et qui imprime le logarithme naturel du sinus de ces nombres en degrés.

Plusieurs solutions sont possibles:

```
Do[Print[Log[Sin[i Degree]]//N], {i,2,5}]

-3.35528
-2.95007
-2.66275
-2.44006
```

Ou bien:

```
For[i=2,i<=5,i++,Print[Log[Sin[i Degree]]//N]]

-3.35528
-2.95007
-2.66275
-2.44006
```

Ou bien:

```
i=2;
While[i<=5,Print[Log[Sin[i Degree]]//N];i++]

-3.35528
-2.95007
-2.66275
-2.44006
```

Ou bien:

```
Table[Log[Sin[i Degree]]//N,{i,2,5}]

{-3.35528, -2.95007, -2.66275, -2.44006}
```

On choisira une procédure plutôt qu'une autre selon le résultat escompté, en ce sens qu'une fonction **Table[ ]** produit une liste, tandis que la fonction **Do[ ]** fait imprimer les résultats. Si la suite des opérations montre que l'enchaînement de la procédure demande une liste, alors **Table[ ]** est préférable à **Do[ ]**. Tout ceci confirme que les procédures ne sont pas nécessairement uniques.

• 2.1.3 Quelques exemples de programmes

Exemple 2.1.3.1.

Ce programme compose une fraction continue. C'est une itération de i = 1 à i = 3:

```
r=x;
Do[r=i/Sqrt[1 - r], {i,1,3}];
r
```

$$\frac{3}{\text{Sqrt}\left[1 - \frac{2}{\text{Sqrt}\left[1 - \frac{1}{\text{Sqrt}[1 - x]}\right]}\right]}$$

Remarque.- Le r de la troisième ligne de la procédure précédente est équivalent à **Return[r]**.

```
??Return

Return[exp] donne la valeur intrinsèque de exp.
Return[ ] donne la valeur Null.

Attributes[Return] = {Protected}
```

Exemple 2.1.3.2.

On peut obtenir quelque chose de semblable avec la fonction **Nest[ ]**

```
Nest[Function[x,1/Sqrt[1 - x]], x, 3]
```

$$\frac{1}{\text{Sqrt}\left[1 - \frac{1}{\text{Sqrt}\left[1 - \frac{1}{\text{Sqrt}[1 - x]}\right]}\right]}$$

Exemple 2.1.3.3.

Voici une itération par *pas* de *deux* (le 2 dans la *liste* {k, 0, 8, 2}])

```
r=2;
Do[r=(r+k)/(1+k), {k,0,8,2}];
r
```

$$\frac{946}{945}$$

Les itérations dans les fonctions **Table[ ]** et **Sum[ ]** se comportent de la même manière que dans **Do[ ]**

```
r=2;
Table[r=(r+k)/(1+k), {k, 0,8,2}]
```

$$\left\{2, \frac{4}{3}, \frac{16}{15}, \frac{106}{105}, \frac{946}{945}\right\}$$

De même

```
r=2;
s=Sum[r=(r+k)/(1+k),{k, 0,8,2}];
{r," ",s}
```

$$\left\{\frac{946}{945}, \ , \frac{6058}{945}\right\}$$

Exemple 2.1.3.4.

Dans l'exécution de la fonction **Table[ ]** qui suit, on remarquera qu'à l'extérieur de la fonction, j a la valeur 1 et dans la fonction, j prend les valeurs de 1 à 4.

```
j=1;
Table[{i,j},{i,3},{j,1+i}]
```

```
{{{1, 1}, {1, 2}}, {{2, 1}, {2, 2}, {2, 3}},
  {{3, 1}, {3, 2}, {3, 3}, {3, 4}}}
```

Les fonctions **Do[ ]** et **Table[ ]** permettent de faire des boucles à l'aide d'une suite de valeurs prises par la variable itérative. Pour plus de flexibilité, il existe dans *Mathematica* les fonctions **For[ ]** et **While[ ]**, elles exécutent une suite d'opérations et s'arrêtent selon certaines conditions impliquant la variable itérative.

Il est essentiel de bien maîtriser toutes ces fonctions pour faire de la programmation. Elles permettent l'exécution d'un groupe d'instructions un certain nombre de fois. Pour un algorithme donné, il est toujours possible d'utiliser l'une ou l'autre de ces structures. Toutefois, l'une d'entre elles peut être plus appropriée qu'une autre, tout dépendra de la suite des opérations.

Exemple 2.1.3.5.

Dans ce qui suit nous ferons une mise en situation impliquant trois fonctions ou procédures dans *Mathematica* :

i) Pour résoudre l'équation n^2 + n - 1 == 0, nous utilisons la fonction **Solve[ ]**

```
Solve[n^2 + n - 1 == 0,{n}]
```

$$\left\{\{n \to \frac{-1 + \text{Sqrt}[5]}{2}\}, \{n \to \frac{-1 - \text{Sqrt}[5]}{2}\}\right\}$$

```
N[n,10]/.%
```

$$\left\{\frac{-1 + \text{Sqrt}[5]}{2}, \frac{-1 - \text{Sqrt}[5]}{2}\right\}$$

```
N[%,10]
{0.6180339887, -1.618033989}
```

ii) On peut obtenir une de ces valeurs (0.6180339887) avec la fonction 1/**FixedPoint[ ]** appliquée à l'expresssion (1+n)/n définie par une fonction pure:

```
1/FixedPoint[(1+#)/#&,1,100]
```

$$\frac{573147844013817084101}{927372692193078999176}$$

```
N[%,15]
0.618033988749895
```

iii) Introduisons la suite de Fibonacci définie par récursivité (les deux premiers éléments de la suite sont égaux à un, on trouve le troisième en additionnant le premier et le deuxième, le quatrième en additionnant le deuxième et le troisième et ainsi de suite...).
Une des procédures qui donne les nombres de Fibonacci est:

```
fibo[1]=fibo[2]=1;
fibo[n_]:=fibo[n]=fibo[n-1]+fibo[n-2]
```

Fin de la procédure.

Voici une liste qui montre les 15 premiers nombres de Fibonacci:

```
Table[fibo[i],{i,15}]
{1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610}
```

Si on va encore plus loin dans la suite de Fibonacci et si on fait le rapport de deux nombres consécutifs, on trouve:

```
N[fibo[100]/fibo[101],15]
0.6180339887498948482
```

L'analyse des parties i), ii) et iii) nous montre que le dénominateur commun de ces trois parties est le nombre `0.6180339887498948482` ou $(-1 + 5^{1/2})/2$ qui est l'inverse du nombre d'Or.

On trouve ce rapport dans divers pentagones, cercles et particulièrement dans le Rectangle d'Or, où il est dit que cette figure est la plus harmonieuse qui soit à l'oeil.

Dans la nature, les étamines du tournesol, la coquille de l'escargot,... se composent selon ce qui s'approche le plus de la spirale de Fibonacci.

Le Parthénon d'Athènes s'inscrit dans un Rectangle d'Or, du moins sa façade.

Le Corbusier a construit plusieurs édifices où les formes extérieures s'inscrivaient dans le Rectangle d'Or. Il disait: «Les Mathématiques exercent une influence harmonieuse sur l'existence».

Piet Mondrian expose, Place de la Concorde, une représentation linéaire de l'art abstrait où il fait appel à des chevauchements de Rectangles d'Or.

Enfin, il semble que Seura s'appuyait dans toutes ses compositions sur ce populaire rapport de grandeur, particulièrement dans les oeuvres suivantes: Le Pont de Courbevoie, Parade et Le Cirque.

Dans tous les calculs faits plus haut, il est important de souligner le nombre d'avenues différentes qui conduisent au nombre d'Or. Le livre de Matila Ghyka, *The Geometry of Art and Life*, est rempli d'exemples tout aussi variés qu'intéressants sur ce qu'est le nombre d'Or.

Exemple 2.1.3.6

Depuis les années '75, avec la découverte des objets fractals par Benoît B. Mandelbrot la géométrie a pris un nouvel essort. Cette branche de la géométrie colle mieux à la nature. Richard F. Voss, confrère et ami de Mandelbrot, se spécialise dans la réalisation de paysages fractals. On retrouve quelques-unes de ses oeuvres dans le film Star Trek II. Peter Oppenheimer est considéré comme un expert de la botanique fractrale. Sa spécialité est les arbres fractals.

Dans les lignes qui vont suivre nous allons réaliser la projection d'arbres fractals par un procédé répétitif qui compose le tronc, les branches et les brindilles. La base consiste à faire une fourchette à trois dents et de la répéter à tous les niveaux de composition des branches. La longueur des éléments qui composent le tronc ou branche ou brindille n'est pas fixe, mais plutôt déterminée par hasard. De même que l'angle entre la branche principale et la branche secondaire est calculé avec la fonction **Random[ ]**, pour plus de réalisme.

Voici le programme qui dessine des «arbres»:

```
Clear[arbre,ligneTriple,exTriples];
rotation = 0.3;
contraction = 0.8;

Needs["Geometry`Rotations`"]

ligneTriple[Line[{s_,d_}]]:=
    {Line[{d,d + contraction Random[Real,{.5,0.8}] Rotate2D[
                  d - s,Random[Real,{.5,1.5}] rotation,{0,0}]}],
    Line[{d,d + Random[Real,{.5,1.5}]contraction(d - s)}],
    Line[{d,d + contraction Random[Real,{.5,0.8}] Rotate2D[
                  d - s,-Random[Real,{.5,1.5}] rotation,{0,0}]}]}

epaisseurcontraction = 0.8;
epaisseuroriginale = 0.05;
```

```
ligneTriple[Thickness[ep_]]:=Thickness[ep epaisseurcontraction];

exTriples[lignes_]:=Flatten[Map[ligneTriple, lignes]]

arbre[profondeur_, rotationR_, contractionC_,
        epaisseurcontractionC_,epaisseuroriginalE_,options___]:=
Block[{},
 rotation = rotationR;
 contraction = contractionC;
 epaisseurcontraction = epaisseurcontractionC;
 epaisseuroriginale = epaisseuroriginalE;
    Graphics[NestList[exTriples,{
     Thickness[epaisseuroriginale],
     Line[{{0,0},{0,1}}]},profondeur]
     ]

]
arbuste=Table[
arbre[
6,
Random[Real,{.3,.5}],
Random[Real,{.750,.850}],
Random[Real,{.60,.80}],
Random[Real,{.02,.03}]
],
{3}]

{-Graphics-, -Graphics-, -Graphics-}
Show[arbuste]
```

```
-Graphics-
```

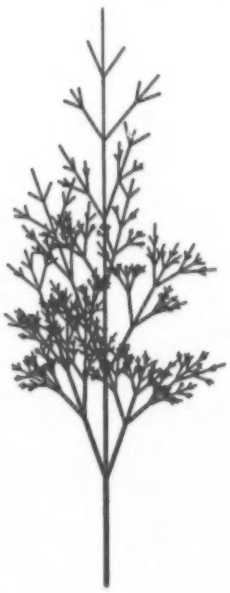
Si nous ne donnons pas plus de précision à la fonction **Show[ ]**, les trois graphiques se superposent tout en gardant l'allure d'arbuste. Ceci illustre le Théorème Collage de Michael F. Barnsley. *La superposition d'objets fractals en 2 D donne encore un objet fractal.*

Parce que les graphiques sont emmagasinés dans une matrice il est toujours possible de les visionner un à la fois.

```
Show[arbuste[[1]],AspectRatio->Automatic]
```

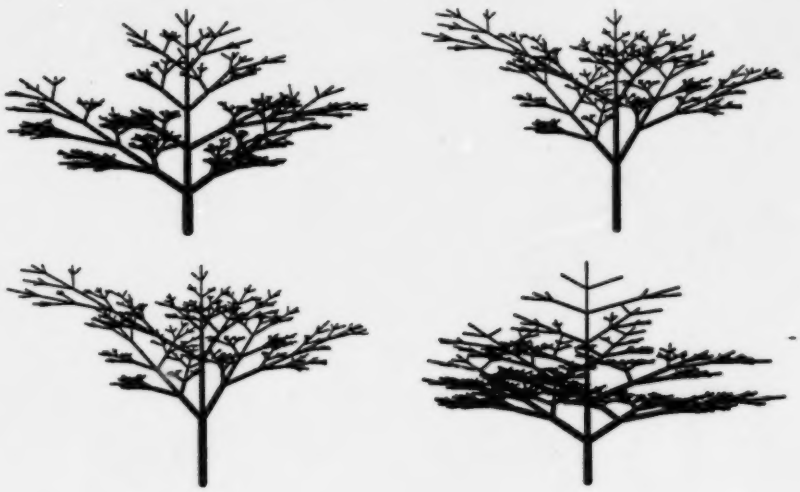
```
Show[arbuste[[2]],AspectRatio->Automatic]
```

```
Show[arbuste[[3]],AspectRatio->Automatic]
```



Mais, grâce à la fonction **GraphicsArray[ ]** nous pouvons voir tous les graphiques sous forme de tableaux

```
Show[GraphicsArray[{{arbuste[[1]],arbuste[[2]]},
{arbuste[[2]],arbuste[[3]]}}]]
```



-GraphicsArray-

□ 2.2 Les procédure avec conditionnelles

Il arrive souvent que pour construire une procédure qui puisse répondre à un problème mathématique donné, on doive se baser sur un arbre de décision, ou mieux un bloc de décision. Il va de soi que pour passer à travers l'arbre ou le bloc de décision il faut tester. On trouve dans *Mathematica* un éventail de fonctions pour faire des tests.

● 2.2.1 Les blocs de décision

Parallèlement aux arbres de décision il existe un autre mode de représentation structurée des algorithmes que nous appelons bloc de décision. Ce sont des graphes emboîtés qui admettent la représentation schématique de la programmation structurée par raffinements successifs et qui permettent de figurer les traitements séquentiels, les tests et les itérations. De nombreux graphes sont faits dans le livre de Pierre Le Beux, *Introduction au PASCAL*.

● 2.2.2 Les procédures avec conditions

Pour faire des procédures il faut connaître les fonctions conditionnelles qui aiguillonnent, comparent, remplacent, vérifient et parfois redonnent la même chose. Un programme composé de sous-programmes ou de sous-routines ou de fonctions a besoin d'un aiguilleur, d'un contrôleur, d'un barreur, d'un agent de décision, etc. Ce sont les conditionnelles qui jouent ce rôle.
Elles sont les principales actrices des programmes à caractère décisionnel.

Voici ce que *Mathematica* contient:

If[*test*, **alors**, autre]	exécute **alors** si *test* est vrai, autre s'il est faux
If[test, alors, autre, *inconnu*]	donne *inconu* si test est ni vrai ni faux. C'est la forme complète de **If**[ ]
Which[*test(1)*, valeur(1), *test(2)*, valeur(2), ...]	évalue les tests tour à tour et donne la valeur associée au premier *test* qui est vrai.
Switch[expression, forme(1), forme(2), ...]	compare expression avec forme(1), forme(2), ... et donne la valeur associée à la première forme qui coïncide avec expression
TrueQ[expression]	retourne **True**, si expression est vrai et **False** sinon
SameQ[expression(1), expression(2)]	retourne **True**, si les deux expressions sont identiques et **False** sinon

Exemple 2.2.1.

Voici une fonction définie par une conditionnelle, si la valeur de x est plus petite ou égale à trois, la fonction retourne huit, sinon, elle retourne moins quatre:

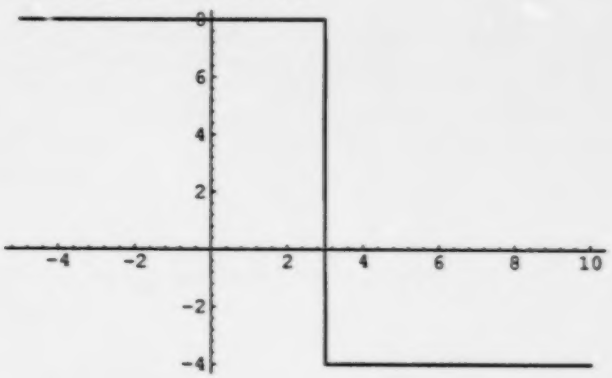
```
f[x_]:= If[x<=3,8,-4]
```

Plaçons dans une liste quelques images de la fonction f[ ]:

```
{f[2],f[0],f[5],f[-3],f[10],f[-2]}
{8, 8, -4, 8, -4, 8}
```

Nous pouvons également tracer le graphe de f[x]:

```
Plot[f[x],{x,-5,10}]
```

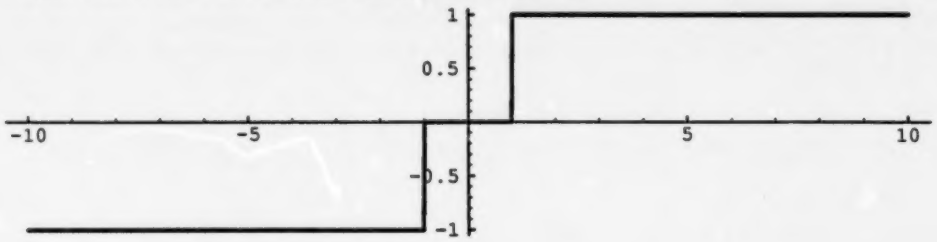


Exemple 2.2.2.

On peut définir une fonction par palier avec la fonction **Which[ ]**:

```
g[x_]:= Which[x<-1,-1,x >-1 && x<1,0, x>1,1]
```

```
Plot[g[x],{x,-10,10},AspectRatio->1/4]
```



Exemple 2.2.3.

Nous allons maintenant définir une fonction qui fait la correspondance entre les chiffres arabes et les chiffres romains.

D'abord vider toutes les variables de leur contenu avec la fonction **Clear[ ]**:

```
Clear[i,v,x,l,c,d,m]
```

La fonction **Switch[ ]** assigne aux lettres i, v, x, l , c, d et m les valeurs des nombres qui correspondent respectivement à 1, 5, 10, 50, 100, 500 et 1000.

```
h[n_]:=Switch[n,i,1,v,5,x,10,l,50,c,100,d,500,m,1000]
```

Aussitôt que la correspondance est effectuée, les opérations sont possibles comme si on calculait des nombres entiers.

Calculer:

```
(h[v]+h[l]-h[c])/-(h[m]+h[v])
```

$$\frac{3}{67}$$

Exemple 2.2.4.

Nous présentons maintenant un petit problème qui prend plutôt l'aspect d'une devinette:

Si on vous dit qu'il faut payer 5$ pour huit, 10$ pour vingt-trois, 35$ pour 204, 45$ pour 1876, 59$ pour 38 902 et 70$ pour 367 792, que pourriez-vous répondre à la question: «Qu'est-ce que le client achète?»

Réponse.- *Des plaques numérologiques!*

Solution. - Le prix est déterminé par le nombre de chiffres dans une plaque, si la plaque contient un chiffre, le coût est de 5$, deux chiffres 10$, trois chiffres 35$, quatre chiffres 45$, cinq chiffres 59$ et six chiffres 70$. Pour établir avec *Mathematica* le prix d'une plaque, on définit la fonction **rigolo[ ]**:

```
rigolo[n_]:=Switch[Length[n],1,5,2,10,3,35,4,45,5,59,6,70]
```

La fonction qui donne le prix total s'appelle **rigolo[ ]** et son argument est une *liste*.
Pour connaître le prix il suffit de passer le numéro de la plaque sous forme de *liste* dans la fonction **rigolo[ ]**:

```
rigolo[{8,3,4,3}]
```

```
45
```

Exemple 2.2.5.

Imaginons une fonction qui n'accepte pas les nombres négatifs, qui exclut tout ce qui se trouve entre 4 et 5 et qui donne un message d'erreur si l'argument de la fonction n'est pas un nombre. De plus, elle multiplie par 5 $ son argument, s'il se trouve dans les conditions qui lui sont imposées.

Pour répondre à ce problème nous allons utiliser la forme complète de **If[ ]**.

```
Clear[f];
f[x_]:=If[x >= 0 &&(x <= 4 || x >= 5),5x" $",
            If[x>0,"Ne corespond pas aux normes de cette
            opération.L'expression doit exclure
            tout ce qui se trouve entre 4 et 5.",
            "Les nombres négatifs ne sont pas acceptés."],
            "Vous vous êtes carrément trompé d'opération!
            Essayez un nombre positif par exemple."]
```

```
f[3^(1/2)]
```

```
"Vous vous êtes carrément trompé d'opération!  Essayez un nombre\
 positif par exemple."
```

```
f[-4]
```

```
Les nombres négatifs ne sont pas acceptés.
```

```
f[a]
```

```
"Vous vous êtes carrément trompé d'opération!  Essayez un nombre\
 positif par exemple."
```

```
f[4.001]
"Ne corespond pas aux normes de cette opération.L'expression doit\
 exclure tout ce qui se trouve entre 4 et 5."
f[8]
40  $
f[0]
0
f[Sqrt[3]//N]
8.66025  $
f[1.73205]
8.66025  $
f[4]
20  $
```

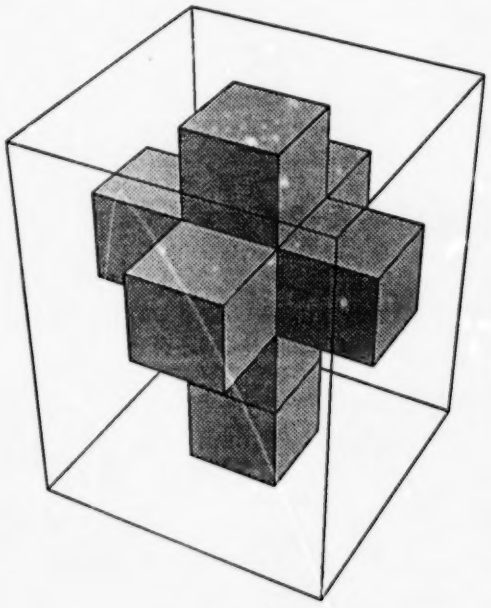
Ordinairement, les expressions d'une procédure dans *Mathematica* sont exécutées à la suite l'une de l'autre. Cependant il peut arriver qu'on veuille changer l'ordre d'exécution, arrêter, bifurquer, revenir, repartir, etc.

Certaines des fonctions sont spécialement adaptées pour ce genre de tâche:

Return[*expression*]	donne la valeur de *expression*
Break[ ]	sort d'une boucle
Continue[ ]	retourne dans la boucle
Goto[*name*]	va à l'élément **Label**[*name*] dans la procédure courante

Troisième partie

La programmation



Il existe plusieurs sortes de programmation: la programmation procédurale, la programmation fonctionnelle et la programmation mathématique.

● 3.1 La programmation procédurale.

Dans la programmation de procédure, les algorithmes sont spécifiés pas à pas. La définition du mot algorithme est donnée par le mathématicien Markov: «Tout ensemble de règles précises qui définit un procédé de calcul destiné à obtenir un résultat déterminé à partir de certaines données initiales.»

Knuth, dans son ouvrage sur l'Art de la programmation ajoute: «C'est un ensemble de règles ayant les cinq caractéristiques suivantes:

- il doit être fini et doit se terminer après un nombre fini d'opérations;
- il doit être défini et précis: chaque instruction doit être définie sans ambiguïté;
- s'il y a des données d'entrée, le champ d'application doit être précisé;
- il doit posséder au moins un résultat;
- il doit être effectif; toutes les opérations doivent pouvoir être effectuées exactement, et dans un temps fini, par un homme utilisant des moyens manuels».

Pierre Le Breux dans son livre *Introduction au PASCAL* donne un avertissement aux jeunes programmeurs: «Il faut cependant avertir le lecteur débutant qu'il *n'existe pas de méthodes* pour découvrir un algorithme concernant un problème non encore résolu, soit personnellement, soit collectivement. La découverte d'un algorithme est donc un acte essentiellement *créatif*, faisant appel non seulement à l'intelligence, mais également à l'intuition, et dans une certaine mesure à l'expérience.» Même au seuil du XXIième siècle aucun ordinateur ne peut faire de tâche créatrice. Leur tâche est programmée par l'homme et est répétitive. Aujourd'hui, on fait faire à l'ordinateur les tâches serviles. Le robot qui travaille et qui peut travailler dans des conditions polaires, s'il est programmé pour faire des soudures, fait uniquement des soudures et pas autre chose. L'ordina - teur qui supervise les robots est probablement programmé sous forme de programmation procédurale.

La programmation procédurale se rencontre dans les langages tels que **C**, **PASCAL**, **FORTRAN** et **BASIC**, pour n'en citer que quelques-uns. À partir d'un organigramme, d'un arbre ou d'un bloc de décision, il est possible d'établir une série de modules qui effectuent une tâche particulière. Les modules sont composés de fonctions telles que **If**, **While**, **For**, **Do**, etc. L'ensemble des modules forme la procédure.

Cependant, il faut chercher à éviter les organigrammes, car ils contiennent des branchements qui s'entrelacent, s'entremêlent, s'enroulent, etc, Ils conduisent souvent à des programmes spaghetti difficiles à déverminer (debug). Il est préférable d'utiliser les blocs de décision qui ont l'avantage d'induire une programmation structurée.

Pour un problème donné, il peut exister plusieurs procédures qui le résolvent, nous devons essayer de trouver celle qui est modulaire. Stephen Wolfram dans *Mathematica, A System for Doing Mathematics by Computer* écrit: «In general, use of **Goto** reduces the degree of structure

that can readily be perceived in a program, and therefore makes the operation of the program more difficult to understand.», et en traduction libre: «En général, l'usage de **Goto** réduit le degré de structure facilement perceptible dans un programme, rendant ainsi son fonctionnement plus difficile à comprendre.»

- **Exemples de programmes basés sur des procédures.**

Exemple 3.1.1

Trouver la racine carrée d'un nombre par la méthode de Newton-Raphson.

Voici le programme basé sur cette méthode qui calcule une racine carrée:

```
Clear[racine,x,x1,z,f,a];
val="donner une valeur près de la racine";
val2="donner une autre valeur près de la racine";
mes="la convergence est très lente, ";
racine[a_]:=Block[{epsilon=10^-10,x,x1,z},
              f[z_]:=z^2 - a;
              x=Input["donner une valeur près de la racine"];
              x1=x;
              While[Abs[f[x1]]>epsilon, x1=x-f[x]/f'[x];x=x1;
              If[f'[x1]>a, Print[mes,val2]];
              Print["la valeur approximative de la racine est "
              ,N[x1,15]]];
              ]
```

Nous allons essayer de calculer la racine carrée de 442.15 en prenant comme valeur près de la racine le nombre 20 (car $20^2 = 400$):

```
racine[422.15]

la valeur approximative de la racine est 20.55375
la valeur approximative de la racine est 20.5462905567719
la valeur approximative de la racine est 20.546289202676
```

La vraie valeur de la racine calculée avec **Sqrt[ ]**:

```
N[Sqrt[422.15],15]

20.546289202676
```

On fait un deuxième essai avec la racine de 13 en prenant 1 comme valeur près de la racine:

```
racine[13]

la convergence est très lente, donner une autre valeur près de la rac
la valeur approximative de la racine est 7.
la valeur approximative de la racine est 4.42857142857143
la valeur approximative de la racine est 3.68202764976959
la valeur approximative de la racine est 3.60634548946552
la valeur approximative de la racine est 3.6055513629176
la valeur approximative de la racine est 3.60555127546399
```

Le programme signale la lenteur de la convergence, mais la valeur devient de plus en plus près de 3.6. La valeur suivante s'enclenche dans l'engrenage du programme et finit par donner la vraie valeur, qui est:

```
N[Sqrt[13],15]

3.60555127546399
```

Commentaires sur les fonctions qui composent le programme:

racine[ ]

la fonction qui calcule la racine.

Block[ ]

fonction utilisée pour donner une valeur locale à certaines variables dont x et epsilon qui prend la valeur 10 à la puissance -10. Cette fonction forme le corps de la procédure. La fonction **Block[ ]** peut être utilisée de deux façons: soit pour définir des variables locales, soit pour donner une valeur particulière à des constantes.

f[z_]:= z^2 - a

fonction utilisée pour extraire la racine carrée de a.

x = Input["...."]

lors de l'exécution du programme, cette fonction amène l'affichage d'une fenêtre demandant la valeur de x. La méthode utilisée, Newton-Raphson, nécessite d'indiquer une valeur qui sert de point de départ; plus cette valeur est près de la racine, plus l'exécution du programme est rapide.

x1 = x

sert de valeur intermédiaire, ce qui permet de repasser à x la valeur après l'exécution du test.

While[ ]

exécute la méthode de Newton-Raphson tant que la précision n'est pas obtenue. Lorsque la valeur absolue de la fonction calculée à x est plus petite qu'epsilon, le résultat est imprimé.

N[x1, 15]

donne la solution avec 15 chiffres significatifs.

Exemple 3.1.2.

Trouver la racine cubique.

Voici un petit programme qui extrait la racine cubique de cinq avec une précision de 40 chiffres significatifs.

```
r=5;x=N[1,40];
Do[x = (2x+r/x/x)/3, {10}];
Print[x]
```

```
1.7099759466766969893531088725438601098 68
```

Vérification:

```
N[5^(1/3),40]==x
```

```
True
```

```
1.7099759466766969893531088725438601098 68^3
```

```
5.
```

Exemple 3.1.3.

Trouver une fonction qui extrait la racine rième de n.

En se basant sur les idées précédentes, on peut écrire une fonction **h[r, n]** qui extrait la racine rième de n.

```
h[r_,n_]:=FixedPoint[((n-1)#+n/#^(r-1))/n&, N[1,25]]
```

La racine 7ième de 8:

```
h[7,8]
```

```
1.34590019263235613194 2833
```

Vérification:

```
N[8^(1/7),25]
1.345900192632356131942833
```

Un deuxième essai avec la racine 20ième de 105:

```
N[105^(1/20),25]
1.26200031977677521337201
h[20,105]
1.26200031977677521337023
```

Commentaires sur les fonctions qui composent le programme:

h[r_,n_]

fonction qui extrait la rième racine de n

FixedPoint[ ]

Exécute l'expression `((n-1)#+n/#^(r-1))/n&` jusqu'à ce le résultat ne change plus.

 # et &

s'utilisent avec des fonctions pures (les fonctions pures seront étudiées dans la section: programmation fonctionnelle).

Exemple 3.1.4.

Imprimer le triangle de Pascal.

Mathematica possède de nombreuses fonctions, il n'est donc pas difficile de trouver l'arrangement de fonctions qui imprime le triangle de Pascal.

```
MatrixForm[Table[{Table[Binomial[n,m],{m,0,n}]},{n,0,13}]]
```

```
{1}
{1, 1}
{1, 2, 1}
{1, 3, 3, 1}
{1, 4, 6, 4, 1}
{1, 5, 10, 10, 5, 1}
{1, 6, 15, 20, 15, 6, 1}
{1, 7, 21, 35, 35, 21, 7, 1}
{1, 8, 28, 56, 70, 56, 28, 8, 1}
{1, 9, 36, 84, 126, 126, 84, 36, 9, 1}
{1, 10, 45, 120, 210, 252, 210, 120, 45, 10, 1}
{1, 11, 55, 165, 330, 462, 462, 330, 165, 55, 11, 1}
{1, 12, 66, 220, 495, 792, 924, 792, 495, 220, 66, 12, 1}
{1, 13, 78, 286, 715, 1287, 1716, 1716, 1287, 715, 286, 78, 13, 1}
```

Commentaires sur les fonctions qui composent le programme:

Binomial[n,m]

le nombre de combinaisons de m objets pris n à la fois

Table[Binomial[n,m], {m,0,n}]

donne les colonnes du triangle

Table[{...},{n,0,13}]

exécute le nombre de lignes du triangle

MatrixForm[ ]

écrit le résultat final sous forme de matrice.

Exemple 3.1.5.

Trouver un programme qui convertit un nombre donné en chiffres romains en chiffres arabes.

Mathematica manipule efficacement les chaînes de caractères, le programme qui suit donne une façon de les mettre en valeur. Dans la section programmation mathématique, nous verrons un programme qui transforme aussi les chiffres romains en chiffres arabes, mais dont l'algoritme est basé sur les règles qui régissent des chiffres romains.

Lors de l'exécution du programme une fenêtre va apparaître demandant: «nombre en chiffres romains?». Le programme s'attend de recevoir une chaîne de caractères, aussi faut-il écrire: **mcmxciii**, par exemple, c'est le format.

```
cr=InputString["chiffre romain?"];
t=0;i=1;v=5;x=10;l=50;c=100;d=500;m=1000;
mm=StringJoin[ToString[cr],"t"];
r=Characters[mm];
n=StringLength[mm];
s1=s2=0;
Do[If[ToExpression[r[[j]]]>=ToExpression[r[[j+1]]],
     s1 = s1+ToExpression[r[[j]]],
     s2 = s2-ToExpression[r[[j]]]],{j,1,n-1}];
Print[Characters[ToString[cr]],"   = ",s1+s2]
```

```
{m, c, m, x, c, i, i, i}   = 1993
```

Commentaires sur les fonctions qui composent le programme:

cr = InputString[ ]

lors de l'exécution du programme, cette fonction amène l'affichage d'une fenêtre demandant la valeur de **cr** sous forme alphanumérique (le format).

t = 0

constante de départ

i = 1; v = 5; ...

donne la correspondance entre les chiffres romains et arabes

StringJoin[ ]

nouvelle suite de caractères alphanumériques caractérisée par la concaténation de **cr** et de **t**

Characters[mm]

liste des caractères dans **mm**

StringLength[mm]

longueur de **mm** ou nombre de caractères

s1 = s2 = 0

valeurs initiales mises à zéro

r[[j]]

représente le $j^{ième}$ caractère de la liste **r**

ToExpression[r[[j]]

donne la valeur numérique de **r[[j]]**

If[ ]

applique les règles des chiffres romains

Do[ ]

exécute les fonctions telles qu'indiquées.

Exemple 3.1.6.

Trouver le programme inverse du programme précédent.

Il va de soi que s'il existe un programme qui transforme les chiffres romains en chiffres arabes, l'inverse existe également.

```
Clear[i,v,x,l,c,d,m]


(* Programme qui convertit des chiffres arabes en romains.
    Ne fait pas la vérification du format.  *)
```

```
str={};
ca=InputString["chiffre arabe; format:0000 < 4 000 ?"];
ra=Characters[ca];
na=StringLength[ca];
rac=Table[ToExpression[ra[[k]]],{k,1,na}];
chrom={m,d,c,l,x,v,i};
dec={"","c, m","x, c","i, x","c, d","x, l","i, v"};
isert[p_,n_]:=Do[AppendTo[str,chrom[[p]]],{n}];
isert[1,rac[[1]]];
     Do[If[rac[[qw]]==9,AppendTo[str,dec[[qw]]],
     If[rac[[qw]] >= 5,isert[2(qw-1),1];isert[2qw-1,rac[[qw]]-5],
     If[rac[[qw]]==4,AppendTo[str,dec[[qw+3]]],
          isert[2qw-1,rac[[qw]]]]]
     ],{qw,2,4}];
Print[ca,"  = ",str]
```

```
2449  = {m, m, c, d, x, l, i, x}
```

Commentaires sur les fonctions qui composent le programme:

(* ... *)

tout ce qui se trouve entre ces deux symboles n'est pas exécuté par *Mathematica*. C'est une des façons de passer les remarques ou commentaires.

str = { }

indique que **str** est une *liste* qui est vide au départ

ca = InputString[ ]

lors de l'exécution du programme, cette fonction amène l'affichage d'une fenêtre demandant la valeur de **ca** sous forme numérique comprise entre 0 et 4000, avec un format de 4 chiffres (e.x.: 1 doit s'écrire 0001)

ra = Characters[ca]

écrit la donnée **ca** sous forme de *liste* dans **ra**

na = StringLength[ca]

donne la longueur de **ca** et le place dans **na**

rac

les caractères de **ra** sont séparés par des virgules pour former une *liste*

chrom

la *liste* des chiffres romains

dec

liste des chiffres romains qui se soustraient, c'est-à-dire qui doivent être lus ensemble, mais en ordre inverse, exemples: 9 (ix), 900 (cm), 4 (iv), etc.

isert

regroupe les lettres des chiffres romains pour former le nombre en chiffres arabes

AppendTo[str,chrom[[p]]]

ajoute le **chrom[[p]]** à la liste **str** et restitue le nouveau **str**

Print[ca," = ", str]

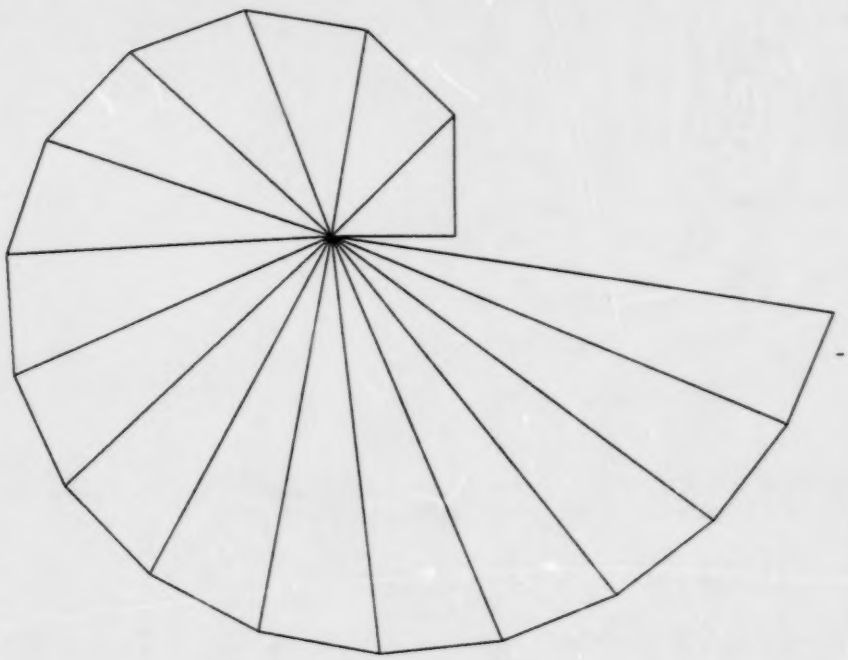
imprime la donnée et le résultat en chiffres romains, sous forme de *liste*.

Exemple 3.1.7.

Un éventail de triangles rectangles dont chaque hypothénuse correspond à la racine de 2, 3, 4, ...

```
Clear[s,r,n,m,l]


unit[r_]:=.25{-r[[2]],r[[1]]}/Sqrt[r[[1]]^2+r[[2]]^2]

m[1]={.25,0};
m[n_]:=m[n]=m[n-1]+unit[m[n-1]]
```

```
sommet=Table[Line[{{0,0},m[n],m[n+1],{0,0}}],{n,1,16}];
Show[Graphics[sommet,AspectRatio->Automatic]]
```



Exemple 3.1.8.

Jeu d'oeufs

Le nombre total d'oeufs d'un certain chargement est tel que lorsqu'on le compte 2 à 2 ou 4 à 4 ou 5 à 5 ou 7 à 7 ou 8 à 8 ou 10 à 10 ou 11 à 11 il reste toujours un oeuf, mais lorsqu'on le compte 13 à 13 il n'en reste aucun. De plus, on veut placer tous les oeufs dans des caisses de 9 par 9 par 9 oeufs. Quel est le nombre minimum de caisses qui satisfait à ces conditions?

Solution:

```
Do[If[IntegerQ[LCM[2, 4, 5, 7, 8, 10, 11] n + 1 / LCM[13, 729]],
   Print[n,    , LCM[2, 4, 5, 7, 8, 10, 11] n + 1]],
  {n, 0, 1000}]


40 123201
```

Le nombre minimum de caisses est:

$$\frac{123201}{729}$$

```
169
```

Exemple 3.1.9.

On lance sur le parquet un brin de spaghetti non cuit. Quelle est la probabilité que, s'il se casse en trois sections, alors ces trois sections puissent former un triangle?

Solution:

RAPPEL:
Random[ ] donne un pseudo-random réel uniformément distribué entre 0 et 1.
Random[**type**,*champ*] donne un nombre pseudo-random selon le type spécifié dans le *champ* demandé. Les **types** possibles sont: Integer (entier), Real (réel) et Complex (complexe). Le *champ* par défaut est entre 0 et 1. Vous pouvez donner le *champ* {min,max} explicitement; une spécification de *champ* max est équivalente à {0,max}.

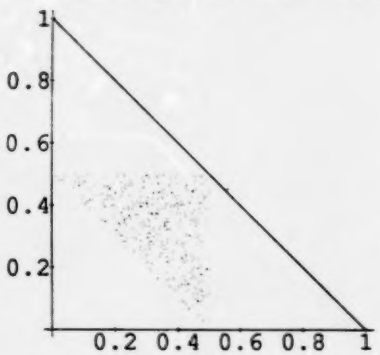
Nous allons faire 3500 essais. On compte les cas favorables divisés par le nombre d'essais. Les cas favorables sont déterminés par le fait que dans un triangle la somme de deux côtés quelconques doit être plus grande que le troisième côté.

```
caspossibles=casfavorables=0;
    Do[x=Random[];y=Random[];
        If[x>0 && y>0 && x+y<1,caspossibles++;
                If[x+y>.5 && x<.5 && y<.5,casfavorables++]],
    {3500}];
Print[casfavorables/caspossibles 100//N," %"]

24.8295 %
```

On arrive à 24.8%.

Avec la méthode dite de Monte-Carlo nous ferons une vérification graphique.

```
s=0;tab={};tab=Table[{x=Random[],y=Random[]};
                                  If[x+y>=.5 && x<=.5 && y<=.5,
                                     {s++;tab[[i]]={x,y}),tab[[i]]={0,0}]
                        ,{i,1,3500}];tab;

p=ListPlot[tab];
l=Graphics[Line[{{0,0},{1,0},{0,1},{0,0}}]];
Show[{p,l},AspectRatio->1]
```



Nous remarquerons que la partie marquée occupe le quart du triangle. Ce qui confirme que la probabilité est de 1/4.

La méthode Monte-Carlo:

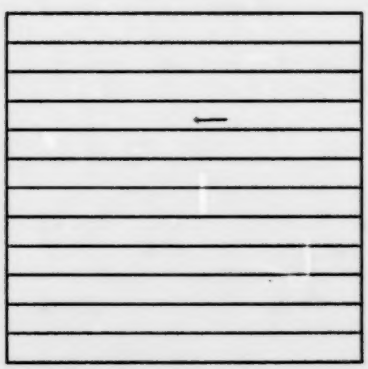
«En arrosant de clous un plancher fait d'étroites lames, on répète en réalité une des expériences, utilisant la technique des probabilités et connue sous le nom de méthode de Monte-Carlo, procédé qui fut déterminant dans la mise au point des blindages des réacteurs atomiques. En multipliant les projections de clous, il est possible de déterminer le nombre de fois où un clou frappera une ligne de séparation entre deux lames. Les savants atomistes adoptèrent cette méthode - la seule à donner des résultats valables et déterminèrent de la sorte les probabilités qu'avait un neutron, issu de la fission d'un noyau atomique, de se voir arrêté dans sa course ou réfléchi par un neutron d'un atome du blindage.» tiré du livre *Les mathématiques, Collection Life*.

Exemple 3.1.10.

Bien avant les expériences connues sous le nom de Monte-Carlo, Buffon (1707-1788) dans son Histoire naturelle raconte l'anecdote de l'aiguille qu'on lance sur le parquet et dont on compte les fois où l'aiguille touche les lignes entre les lames du parquet. Les cas favorables et les cas non-favorables déterminaient le nombre π. Cependant, on doit ajouter que les lames du parquet devaient avoir une largeur précisement égale à la longueur de l'aiguille.

On dit qu'à la Faculté des Sciences de Paris, on expose une machine qui compte les cas favorables et les essais. Plus le nombre d'essais est grand plus le rapport se rapproche de π. Voici un programme qui fait office de machine. Non seulement il compte les cas favorables des essais, mais il dessine l'aiguille et le parquet.

```
p=3 Input["Grandeur du parquet carré; taper un nombre
représentant le côté d'un parquet en mètres"];
pi=N[Pi,20];fav=0;
h[_]:=Random[Real,{-p pi,p pi}];

k[_]:=Random[Real,{-(p-1),(p-1)}];

rot[theta_]:={{Cos[theta],Sin[theta]},{-Sin[theta],Cos[theta]}};


essais=Input["Le nombre d'essais?"];

Do[
dep1=k[1];dep2=k[1];
{m,n}={0,.5}.rot[h[1]];
{{a,b},{c,d}}={{dep1,dep2},{2 m + dep1,2 n + dep2}};

     aig=Line[{{a,b},{c,d}}];
     con=Line[{{p,p},{-p,p},{-p,-p},{p,-p},{p,p}}];
     par=Table[Line[{{-p,k},{p,k}}],{k,-(p-1),(p-1)}];
     Show[Graphics[{{RGBColor[1,.1,0],aig},
     {RGBColor[.1,1,.1],PointSize[0.015],Point[{c,d}]},con,
     {RGBColor[.1,.2,1],par}}],
     AspectRatio->1];
     Print[Sqrt[(c-a)^2+(d-b)^2]];


Do[If[(y>d && y<b)||(y>b && y<d),fav++],{y,-(p-1),(p-1)}];
,{essais}];
Print["Après ",essais," essais on arrive à ",
     2 fav/(essais-fav)//N,
     " sur un parquet de ",p/3,If[p/3 <=1,
     " mètre de côté."," mètres de côté."]]
```



On doit enlever la partie qui dessine le parquet si nous voulons faire un grand nombre d'essais. Nous avons fait tourner le programme avec un parquet de deux mètres sur deux mètres et avec dix mille essais.

```
p=3 Input["Grandeur du parquet carré; taper un nombre
représentant le côté d'un parquet en mètres"];
pi=N[Pi,20];fav=0;
h[_]:=Random[Real,{-p pi,p pi}];

k[_]:=Random[Real,{-(p-1),(p-1)}];

rot[theta_]:={{Cos[theta],Sin[theta]},{-Sin[theta],Cos[theta]}};


essais=Input["Le nombre d'essais?"];

Do[
dep1=k[1];dep2=k[1];
{m,n}={0,.5}.rot[h[1]];
{{a,b},{c,d}}={{dep1,dep2},{2 m + dep1,2 n + dep2}};


Do[If[(y>d && y<b)||(y>b && y<d),fav++],{y,-(p-1),(p-1)}];
,{essais}];
     Print["Après ",essais," essais on arrive à ",,
     2 fav/(essais-fav)//N,
     " sur un parquet de  ",p/3,If[p/3 <=1,
     " mètre de côté."," mètres de côté."]]

Après 10000 essais on arrive à 3.141504 sur un parquet de 2

   mètres de côté.
```

La programmation procédurale, tout en étant efficace, n'a rien d'esthétique et n'est pas facile à manipuler. Il en est tout autrement de la programmation fonctionnelle.

□ 3.2 La programmation fonctionnelle

Les relations fonctionnelles.

Dans *Mathematica* le terme fonction a le sens de *relation fonctionnelle* dans certains cas, et le sens d'*opérateur mathématique*, dans d'autre cas.

Les deux exemples suivants montrent quel est le rôle d'une fonction dans *Mathematica*.

Exemple 3.2.1.

La fonction ci-dessous représente une fonction cube où on effectue une retenue de 2 % sur l'argument au carré et de 4 % sur l'argument lui-même.

```
rcube[x_]:=x^3 - x^2 .02 - x .04
```

Si l'argument est 5, on obtient:

```
rcube[5]
124.3
```

Vérification:

```
125 - 25 2/100 - 5 4/100//N
124.3
```

Si on doit résoudre un problème où le nom de la fonction évoque la solution ou contribue d'une certaine façon à rendre plus compréhensible son utilisation, c'est le genre de fonction qu'il faut utiliser.

Cependant, si le nom de la fonction est sans importance et si ce qui compte est l'arrangement des variables, alors dans ce cas on peut utiliser une *fonction pure*.

Voici un exemple d'une fonction pure:

```
Function[x, x^3 - x^2 .02 - x .04] [5]
124.3
```

Le nom de la fonction n'est pas impliqué et le résultat est identique.

On peut même ajouter que l'argument x ne joue qu'un rôle de position, qu'il s'appelle x ou y ou z

n'a pas tellement d'importance. Dans ce cas *Mathematica* utilise une variable muette, représentée par: #

```
Function[#^3 - #^2 .02 - # .04] [5]

124.3
```

Comme les mathématiques rivalisent avec la sténographie, le terme **Function** peut être remplacé par &, d'où une *fonction pure*.

```
#^3 - #^2 .02 - # .04 & [5]

124.3

??&

Function[corps] ou corps& est une fonction pure. Les paramè-
tres sont # (ou #1), #2, etc. Function[x, corps] est une
fonction pure utilisant le paramètre singulier x .
Function[{x1, x2, ...}, corps] est une fonction pure avec
une liste de paramètres formels.
Function[{x1, x2, ...}, corps,{attributs}] conserve ses attributs
durant l'évaluation.

Attributes[Function] = {HoldAll, Protected}
```

Remarque.- Le perluète & remplace le mot **Function[ ]** dans l'expression de la fonction et se place après la description des variables.

Exemple 3.2.2.

Soit la fonction pure:

```
Expand[#^5&'[a+b]]/.a->1

5 + 20 b + 30 b^2 + 20 b^3 + 5 b^4
```

Voici une procédure qui fait la même tâche que la fonction précédente:

```
f[x_]:=x^5;  (* une fonction monomiale du 5ième degré *)

f'[a+b];  (* la dérivée au point a+b *)

Expand[%];  (* développer le résultat précédent *)

%/.a->1  (* dans ce qui précède remplacer a par 1 *)


5 + 20 b + 30 b^2 + 20 b^3 + 5 b^4
```

Comme on peut le constater l'utilisation d'une fonction pure réduit considérablement l'écriture. La lecture d'une fonction pure est beaucoup plus dificile à comprendre qu'une fonction détaillée.

Et que devons-nous penser de l'expression suivante?:

```
Expand[#1^5&'[#2+#3]]/.#2->1/.#3->b
```

$$5 + 20\,b + 30\,b^2 + 20\,b^3 + 5\,b^4$$

La section 2.2.5. du livre de *Mathematica* est consacrée aux fonctions pures.

Exemple 3.2.3.

D'un vecteur trouver les cosinus directeurs en degrés, minutes et secondes.

Les idées qui suivent ne sont pas nouvelles, elle ont déjà fait l'objet de recherche dans des petits programmes faits en programmation procédurale. Nous avons regroupé les différentes procédures pour arriver finalement à **trans[a_,b_,c_]**: la fonction qui, à partir d'un vecteur, donne les cosinus directeurs en degrés, minutes et secondes.

```
trans[a_,b_,c_]:=
    Block[{x,y,z,i},
        sf[x_,y_,z_]:=N[ArcCos[{x,y,z}/({x,y,z}/.{x,y,z}->
            Sqrt[x^2+y^2+z^2])] 180/Pi,15];
        dms[x_]:=(int=(x-Floor[x])60;
            degre=Floor[x];
            minute=Floor[int];
            seconde=(int-Floor[int]) 60;
            Print[degre, "° ", minute, "' ", seconde, " '' "]);
        Do[dms[sf[a,b,c][[i]]],{i,1,3}]
    ]
```

Trouver les cosinus directeurs du vecteur (7, 8, 9):

```
trans[7,8,9]
```

```
59° 49' 47.1968 ''
54° 56' 40.9857 ''
49° 44' 49.6321 ''
```

Remarque.- Nous avons repris cette procédure pour montrer que la cloison entre la programmation procédurale et la programmation fonctionnelle est bien mince.

Exemple 3.2.4.

Écrire un programme fonctionnel qui donne les racines réelles ou complexes d'un polynôme du premier ou du second degré.

L'algorithme, basé sur un bloc-décision, s'identifie avec le programme structuré ci-dessous:

```
racines::usage = "donne les racines d'un polynôme du 1° degré
et du 2° degré.  ";
    simp::= Simplify;
    racines[a_,b_,c_]:= Block[{delta = b^2 - 4 a c},
        If[a==0,Print["premier degré"];
            If[b==0,Print["solution dégénérée"];
                If[c==0,Print["indéterminé"],Print["impossible"]]
                        ,Print["une solution"];
                            x=simp[-c/b];
                            Print[x]]
                    ,Print["second degré"];
                If[delta==0,Print["racine double"];
                    xx=simp[-b/2 /a];
                    Print[xx],
                If[delta >0,Print["deux racines réelles"];
                    x1 = simp[(-b+Sqrt[delta])/2 /a];
                    x2 = simp[(-b-Sqrt[delta])/2 /a];
                    Print[x1,"     ",x2];
            ,Print["deux racines complexes"];
                x3 = simp[(-b+Sqrt[delta])/2/a];
                x4 = simp[(-b-Sqrt[delta])/2/a];
                Print[x3,"      ",x4]
            ]]]
                ]
```

Remarque.- La lecture et l'observation du programme donne une bonne indication de sa construction qui est une suite de conditionnelles emboîtées.

Un tel programme se comporte dans *Mathematica* comme une véritable fonction. Cette fonction est à trois variables où chacune des variables représente les coefficients des puissances d'une équation quadratique.

Pour trouver les racines de 3 x^2 + 4 x+ 5 == 0, par exemple, on écrit :

```
racines[3,4,5]

second degré
deux racines complexes
-4 + 2 I Sqrt[11]      -4 - 2 I Sqrt[11]
-----------------      -----------------
        6                      6
```

Voici quelques essais pour étudier le comportement de la fonction:

racines[1,1,1]

```
second degré
deux racines complexes
-1 + I Sqrt[3]      -1 - I Sqrt[3]
--------------      --------------
      2                   2
```

racines[1,1,-1]

```
second degré
deux racines réelles
-1 + Sqrt[5]      -(1 + Sqrt[5])
------------      --------------
     2                 2
```

racines[0,2,3]

```
premier degré
une solution
  3
-(-)
  2
```

racines[1,2,1]

```
second degré
racine double
-1
```

racines[0,0,1]

```
premier degré
solution dégénérée
impossible
```

racines[9,23,61]

```
second degré
deux racines complexes
-23 + I Sqrt[1667]      -23 - I Sqrt[1667]
------------------      ------------------
        18                      18
```

Commentaires sur les fonctions qui composent le programme:

If[ ]

fonction conditionnelle agissant comme filtre

racines :: usage = ...

décrit ce que fait la fonction que l'on nomme **racines**

delta

est le discriminant $b^2 - 4\,a\,c$

Simplify[ ]

écrit une fraction sous forme simplifiée

simp :: = Simplify

assemblage qui permet de remplacer une fonction de *Mathematica* par un mot plus simple à utiliser.

Exemple 3.2.5.

Trouver un moyen de placer huit dames sur un échiquier sans qu'elles soient en prise.

Aux échecs, la dame est la pièce maîtresse, elle contrôle toutes les cases de la ligne, de la colonne et des deux diagonales qui la contiennent. Le jeu consiste à faire coexister huit dames sur un échiquier sans qu'elles soient en prise en aucun moment.

Le but de cet exemple est d'écrire un programme qui donne toutes les positions possibles de coexistence.

En jouant sur un échiquier avec huit pions, convertis en dames pour les besoins du jeu, il est possible de découvrir l'algorithme permettant d'arriver à la solution.

Le programme qui suit nous donne toutes les solutions; il y a en tout 92 configurations de huit dames sur l'échiquier. Le 92 est trouvé par énumération et non par démonstration. Est-il possible de faire ce genre de démonstration?

Avant d'écrire le programme, regardons les fonctions les plus importantes:

Le rôle de la fonction **rep[ ]** est de recueillir les solutions numérotées en commencant par un et de les imprimer sous forme de tableau. Elle indique une positon favorable pour une dame par un * et une positon défavorable par un o.

Le rôle de **oqp[,]** est de vérifier si aucune ligne ou diagonale n'entrent en conflit. On se base sur le fait que s'il y a coïncidence sur une diagonale, la pente en valeur absolue est égale à un. La fonction **oqp[,]** est booléenne, son résultat est une *liste* contenant des **True** et des **False**. Dans le programme principal, la fonction **Count[ ]** dénombre la fréquence de **True** dans *liste*; si au moins un **True** est trouvé, il y a conflit et on entreprend la recherche d'une autre position.

Le programme principal balaie l'échiquier sur les huit colonnes et revient en arrière s'il n'a pas trouvé de solution (back-tracking). Quand une solution complète est trouvée, il l'imprime avec **rep[ ]** et il analyse la solution suivante. Il travaille ainsi jusqu'à ce que toutes les solutions soient trouvées.

Les fonctions **oqp[,]** et **rep[ ]** sont en fait des sous-routines qui sont appelées par le programme principal. Ce sont des sous-routines fonctionnelles où on passe les variables locales par les arguments des fonctions lorsque nécessaire.

Voici le programme:

```
Print["Il faut quelques minutes au programme pour trouver la première solution."];

n=8;arr={,,,,,,,};sol=0;
Clear[oqp,rep,i,j,k,s,c,col,ncol,colonne,ligne,mat];
oqp[c_,l_]:= (Table[col=Abs[arr[[k]]-l];If[col*(col-c-1+k)==0,
                        True,False],{k,1,c}])

rep[solno_]:=(sol++;
              mat=Table[If[j==arr[[i]],"*","°"],{i,1,n},{j,1,n}];
              Print["sol no ",sol];
              Print[" "];
              Print[TableForm[mat]];
              Print[" "])
nsol=0;
Do[arr[[1]]=s;colonne=1;ligne=1;
```

```
While[colonne!=0,

        While[ligne <= 8,(If[Count[oqp[colonne,ligne],True]!=0,
                    ligne++,
                        (arr[[colonne + 1]] = ligne;
                            colonne ++;
                                If[colonne <= n - 1,
                                    ligne = 1,
                                    (rep[solno];
                                    ligne=n+1)])
        ])
];
    While[ligne >8 && colonne != 0,
        (colonne--;
        ligne = arr[[colonne+1]]+1)
    ];
],
{s,1,n}]
```

Il faut quelques minutes au programme pour trouver la
première solution.

sol no 1

```
*  o  o  o  o  o  o  o
o  o  o  o  *  o  o  o
o  o  o  o  o  o  o  *
o  o  o  o  o  *  o  o
o  o  *  o  o  o  o  o
o  o  o  o  o  o  *  o
o  *  o  o  o  o  o  o
o  o  o  *  o  o  o  o
```

sol no 2

```
* o o o o o o o
o o o o o * o o
o o o o o o o *
o o * o o o o o
o o o o o o * o
o o o * o o o o
o * o o o o o o
o o o o * o o o
```

sol no 3

```
* o o o o o o o
o o o o o o * o
o o o * o o o o
o o o o o * o o
o o o o o o o *
o * o o o o o o
o o o o * o o o
o o * o o o o o
```

sol no 4

```
* o o o o o o o
o o o o o o * o
o o o o * o o o
o o o o o o o *
o * o o o o o o
o o o * o o o o
o o o o o * o o
o o * o o o o o
```

Il y a en tout 92 solutions, si on considére un échiquier classique. La première et la 92ième solution, la deuxième et la 91ième, ... sont symétriques, soit par rotation, soit par réflexion. Il serait intéressant de faire un programme qui ne considérerait que les solutions asymétriques. Combien y aurait-il de solutions vraiment différentes?

Exemple 3.2.6.

Écrire un programme qui trace la spirale de Fibonacci.

On a déjà vu en programmation procédurale une façon d'obtenir la suite de Fibonacci, en voici une autre.

```
fibo[q_Integer?Positive]:=Block[{r=1,p=0},
                               Do[{r,p}={r+p,r},{q-2}];
                               p+r]
```

Formons une *liste* des 15 premiers nombres de Fibonacci:

```
Table[fibo[i],{i,1,15}]
```

```
{1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610}
```

La fonction **fibo[ ]** vérifie si l'argument est bien un nombre entier positif. Avec un argument négatif ou nul *Mathematica* n'a aucune définition concernant **fibo[ ]**.

```
fibo[-15]
```

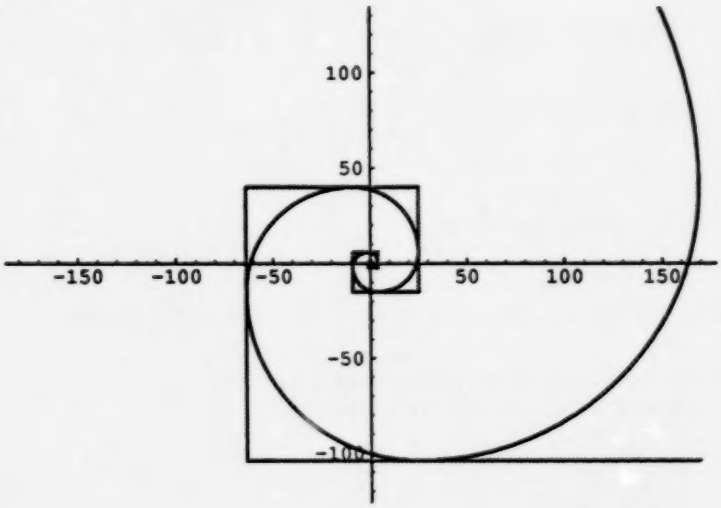
```
fibo[-15]
```

Voici le programme qui trace la spirale:

```
Clear[a,b,abs,ord,m,g,k,r,p,min,max,sp,cerdes,spiral];
abs[k_]:=If[OddQ[k],Sum[(-1)^((i+1)/2+1) fibo[i],
        {i,k-2,1,-2}],abs[k-1]]
ord[k_]:=If[EvenQ[k],Sum[(-1)^(i/2+1) fibo[i],
        {i,k-2,2,-2}],ord[k-1]]
m = Table[{abs[j],ord[j],fibo[j],Mod[j,4]},{j,1,15}];
cerdes[{a ,b ,r ,p }]:={a + r Cos[x],b + r Sin[x]}
```

```
min[t_]:=Switch[t,1,0,2,Pi/2,3,Pi,0,3 Pi/2];
max[t_]:=Switch[t,1,Pi/2,2,Pi,3,3 Pi/2,0,2 Pi];
spiral:=Table[ParametricPlot[cerdes[m[[sp]]],
{x,min[m[[sp]][[4]]],max[m[[sp]][[4]]]},
AspectRatio->Automatic],{sp,1,15}];
ssm=Table[{m[[g]][[1]],m[[g]][[2]]},{g,1,Length[m]}];
gssm=Graphics[{RGBColor[1,0,1],Line[ssm]}];
Show[spiral,gssm]
```

Finalement la spirale de Fibonacci:



Remarque.- Dans la spirale d'Archimède, le centre est fixe et le rayon augmente proportionnellement à l'angle polaire, alors que dans la spirale de Fibonacci, le centre se déplace aux sommets de carrés dont les côtés sont formés par les nombres de la suite de Fibonacci.

Commentaires sur les fonctions qui composent le programme:

Clear[ ]

vide tous les arguments de cette fonction

abs[k_], ord[k_]

donnent respectivement l'abscisse et l'ordonnée des centres des cercles

cerdes[{a_,b_,r_,p_}]

fonction qui décrit un arc de cercle de centre (**a**, **b**) et de rayon **r**, **p** est une variable muette, c'est-à-dire qu'elle ne joue aucun rôle spécial sauf un rôle de position.

min[ ], max[ ]

fonctions qui donnent les minimums et les maximums de la **Table[ ]** créée par **spiral**

spiral

table qui contient tous les arcs de cercle qui forment la spirale de Fibonacci

ssm

tableau qui contient les sommets des carrés qui circonscrivent la spirale

gssm

transforme en graphique la fonction **Line[ ]** qui relie les sommets de **ssm** et crée les carrés

Show[ ]

fonction qui montre la spirale et les carrés qui soutiennent la spirale de Fibonacci.

Exemple 3.2.7

Écrire un programme qui donne les hypercubes de côté un.

D'une façon générale, pour caractériser une composante à **k** dimensions (*on entend par composante les sommets, les arêtes, les faces, les cubes, etc.*) dans un espace à **n** dimensions, il faut d'abord choisir parmi **n** coordonnées celles qui forment les côtés, *i.e. le nombre de combinaisons de k objets choisis parmi n*, et ensuite choisir parmi les **n - k** coordonnées qui restent celles qui doivent être soit 0 ou soit 1, *i.e.* 2^{n-k}. Le nombre de choix possibles en tout est le produit de: *le nombre de combinaisons de k objets choisis parmi n* multiplié par 2^{n-k}.

Avec *Mathematica* pour faire un programme qui nous donne les composantes des hypercubes, il suffit de multiplier les coefficients du binôme de Newton par l'exponentielle de base 2 où l'argument est (n-m).

```
MatrixForm[Table[{{n, " D "},Table[Binomial[n,m] 2^(n-m),
{m,0,n}]},{n,0,6}]
]
```

```
{0,  D }                          {1}
{1,  D }                          {2, 1}
{2,  D }                          {4, 4, 1}
{3,  D }                          {8, 12, 6, 1}
{4,  D }                          {16, 32, 24, 8, 1}
{5,  D }                          {32, 80, 80, 40, 10, 1}
{6,  D }                          {64, 192, 240, 160, 60, 12, 1}
```

Si on se réfère à la ligne {3, D} on peut lire la liste {8, 12, 6, 1}, qui doit être interprétée comme suit: dans un cube en trois dimensions il y a 8 sommets, 12 arêtes, 6 faces et 1 cube 3D. L'interprétation de la ligne suivante est: dans un hypercube en quatre dimensions il y a: 16 sommets, 32 arêtes, 24 faces, 8 cubes 3D et 1 hypercube 4D.

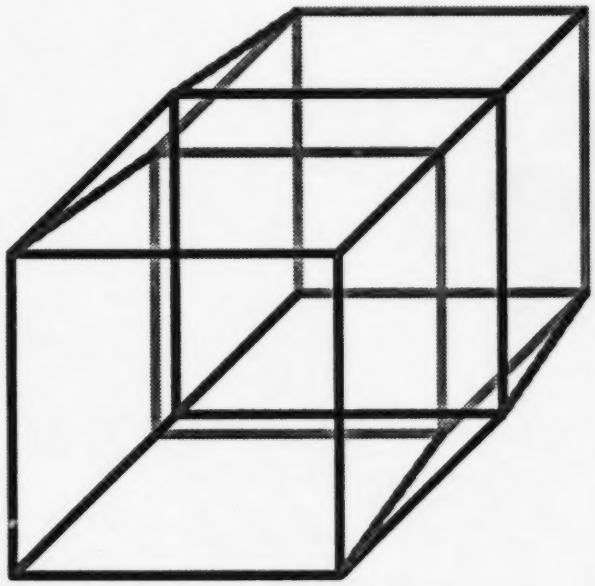
Évidemment on ne peut pas se faire une représentaion géométrique d'un hypercube en 4Ð, mais par analogie avec le fait qu'un cube en 3D a une projection 2D formée de parallélogrammes, la projection d'un hypercube 4D en 3D est formée de paralilélépipèdes. Voici un programme qui donne cette projection:

```
hyperCube[a_,b_,c_,d_,sa_,sb_,sc_,sd_,w_]:=(r=Line[{{b,b},{d,b},
{d,d},{b,d},{b,b}}];
r1=Line[{{a,a},{c,a},{c,c},{a,c},{a,a}}];
r2=Line[{{a,a},{b,b}}];
r3=Line[{{a,c},{b,d}}];
r4=Line[{{c,c},{d,d}}];
r5=Line[{{c,a},{d,b}}];
sr=Line[{{sb,sb},{sd,sb},{sd,sd},{sb,sd},{sb,sb}} w];
sr1=Line[{{sa,sa},{sc,sa},{sc,sc},{sa,sc},{sa,sa}} w];
sr2=Line[{{sa,sa},{sb,sb}} w];
sr3=Line[{{sa,sc},{sb,sd}} w];
sr4=Line[{{sc,sc},{sd,sd}} w];
sr5=Line[{{sc,sa},{sd,sb}} w];
hc=Line[{{a,a},{sa,sa} w}];
hc2=Line[{{b,b},{sb,sb} w}];
hc3=Line[{{c,c},{sc,sc} w}];
hc4=Line[{{d,d},{sd,sd} w}];
hc5=Line[{{b,d},{sb,sd} w}];
hc6=Line[{{d,b},{sd,sb} w}];
hc7=Line[{{a,c},{sa,sc} w}];
hc8=Line[{{c,a},{sc,sa} w}];
Show[Graphics[{Thickness[0.015],RGBColor[0,1,.3],r,r1,r2,r3,r4,r5,
RGBColor[.2,0,1],sr,
sr1,sr2,sr3,sr4,sr5,
RGBColor[1,.1,.2],hc,hc2,
hc3,hc4,hc5,hc6,hc7,hc8}],
AspectRatio->Automatic])
```

```
Table[hyperCube[1,2,3,4,0,4,8,12,x/5],{x,.325,3.5,.0625}]
```

La **Table[ ]** qui précède forme une série de graphiques pour l'animation. Le cas particulier qui suit montre une des projections.

```
hyperCube[1,2,3,4,0,4,8,12,.285]
```

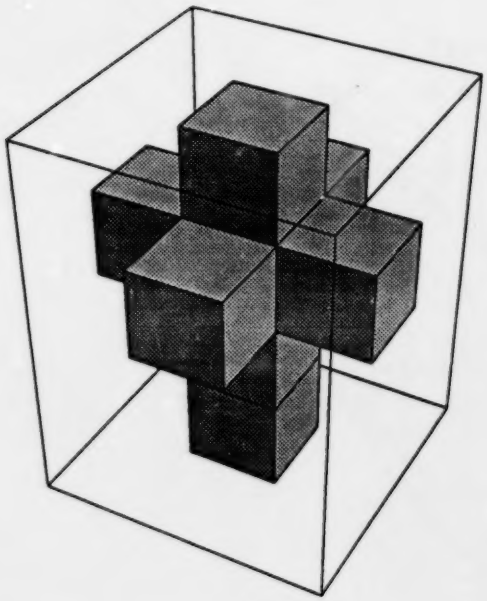


```
-Graphics-
```

Un cube 3D se développe en 2D en six carrés, un hypercube 4D se développe en 3D en huit cubes:

```
Show[Graphics3D[
     {Cuboid[{1,1,1}],Cuboid[{1,1,2}],
     Cuboid[{1,1,0}],Cuboid[{0,1,1}],Cuboid[{1,0,1}],
```

```
Cuboid[{2,1,1}],Cuboid[{1,2,1}],
Cuboid[{1,1,-1}]
}
]]
```



Ce sont ces huit cubes qui se déforment pour donner les parallélépipèdes de la projection 3D de l'hypercube.

Salvador Dali a peint un Christ crucifié sur le développement d'un hypercube et l'ombre de ce dernier forme sur le plancher une croix grecque, qui, elle-même, est le développement d'un cube ouvert. Cette croix n'est pas là par hasard puisqu'elle vient briser la monotonie du dallage en damier noir et blanc. Cf. The Crucifixion, The Metropolitan Museum of Art, Chester Dale Collection.

Exemple 3.2.8.

Écrire un programme qui donne la somme des nombres naturels élévés à n'importe quelle puissance.

Lorsque nous calculons une intégrale selon Riemann il nous arrive d'être confronté à des limites de la forme:

```
Limit[1^2+2^2+3^2+...+(n-1)^2+n^2/n^3,n->Infinity]
```

Evidement, *Mathematica* ne peut pas calculer ce genre de limites, aussi est-il préférable de remplacer le numérateur, la somme des carrés des nombres natures, par la formule:

```
n(1+n)(1+2n)/6
```

ce qui permet de simplifier la limite. Cette simplification peut être prise par *Mathematica*:

```
Limit[n(1+n)(1+2n)/6/n^3,n->Infinity]
```

$$\frac{1}{3}$$

Voici le programme qui écrit la formule de la somme des nombres naturels élevés à n'importe quelle puissance:

```
BeginPackage["Sommation`"]
SommationPuis ::usage = "...donne la somme des n premiers entiers exposant x"
Begin["Private`"]
SommationPuis[x_] :=
 Block[{i,j,m,ind,coef,poly},
  r = x + 1;
  m = Table[i^(r-j),{i,r},{j,r}];
  ind = Table[Sum[i^(r-1),{i,j}]/j,{j,r}];
  coef = Inverse[m].ind;
  poly = Table[n^(r+1-j),{j,r}];
  Factor[poly.coef]
 ]
End[]
EndPackage[]

pt[m_]:=Print["1^",m," + 2^",m," +3^",m," +...","+ n^",m," ="];
Ps[m_]:=(pt[m];SommationPuis[m]);
```

Pour obtenir la somme des n premiers entiers au carré, il suffit d'écrire **Ps[2]**, au cube **Ps[3]**, etc.

Ps[2]

$$1^2 + 2^2 + 3^2 + \ldots + n^2 = \frac{n\,(1 + n)\,(1 + 2\,n)}{6}$$

Ps[3]

$$1^3 + 2^3 + 3^3 + \ldots + n^3 = \frac{n^2\,(1 + n)^2}{4}$$

Et finalement, la formule qui donne la somme des n premiers nombres naturels élevés à la puissance 15:

Ps[15]

$$1^{15} + 2^{15} + 3^{15} + \ldots + n^{15} = \frac{n^2\,(1 + n)^2\,(420 - 840\,n - 122\,n^2 + 1084\,n^3 - 226\,n^4 - 632\,n^5 + 203\,n^6 + 226\,n^7 - 83\,n^8 - 60\,n^9 + 21\,n^{10} + 18\,n^{11} + 3\,n^{12})}{48}$$

Exemple 3.2.9.

Faire un programme qui montre que l'aire d'un cercle peut être considérée comme la limite de polygones inscrits dans le cercle lorsque le nombre de polygones tend vers l'infini.

Le programme qui suit fait apparaître un cercle où s'inscrivent des triangles isocèles égaux. Ces triangles forment les polygones.

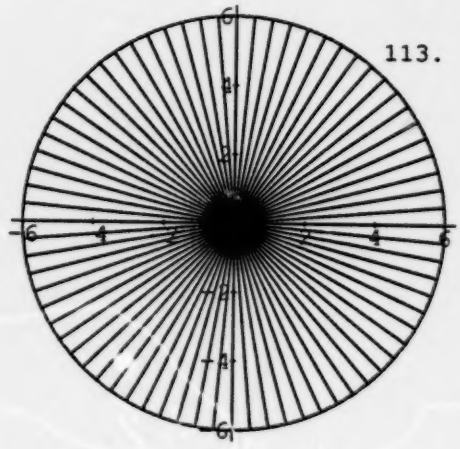
Le programme fait la somme des aires des triangles. Ce résultat est ensuite comparé avec la valeur réelle de l'aire du cercle.

La caractéristique principale de ce programme est qu'il peut s'animer, il suffit de cliquer le rectangle qui se situe à gauche de la cellule et de taper simultanément la touche contrôle et la lettre y.

Il y a en tout près d'une vingtaine de graphiques dont l'aire tend finalement vers le nombre 113.097.

Ce graphique peut être animé.

```
Clear[t,r,k,m,sect,pf,f]
f[x_]:= N[36 2 x Sin[Pi/2/x],4]
gt=Table[f[q],{q,1,18}]
tt=Table[Graphics[Text[gt[[r]],{5,5}]],{r,1,18}]
r[t_]:= 6
pf=ParametricPlot[{r[t] Cos[t],r[t] Sin[t]},{t,0, 2Pi},
AspectRatio->Automatic,PlotStyle->RGBColor[1,0,.1]]
sect[k_]:=Graphics[Line[{{0,0},{6 Cos[ k Pi/m],6 Sin[ k Pi/m]},
{6 Cos[ (k-1) Pi/m], 6 Sin[ (k-1) Pi/m]},{0,0}}]]
Do[Show[Table[sect[r],{r,1,2m}],pf,tt[[m/2]]],
{m,2,36,2}]
Print["La somme des aires des triangles tend vers  113.097"]
```



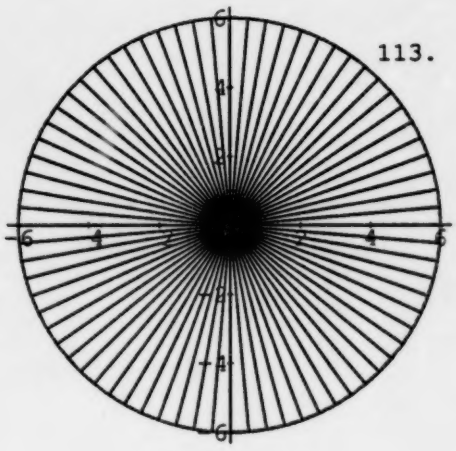

```
La somme des a        s triangles tend vers  113.097
```

Exemple 3.2.10.

Faire un programme qui trace deux courbes de fonctions et qui dessine des rectangles dont la somme des aires calcule l'aire entre les courbes.

Ce graphique peut être animé.

```
Clear[t,r,k,m,sect,pf,f]
f[x_]:= N[36 2 x Sin[Pi/2/x],4]
gt=Table[f[q],{q,1,18}]
tt=Table[Graphics[Text[gt[[r]],{5,5}]],{r,1,18}]
r[t_]:= 6
pf=ParametricPlot[{r[t] Cos[t],r[t] Sin[t]},{t,0, 2Pi},
AspectRatio->Automatic,PlotStyle->RGBColor[1,0,.1]]
sect[k_]:=Graphics[Line[{{0,0},{6 Cos[ k Pi/m],6 Sin[ k Pi/m]},
{6 Cos[ (k-1) Pi/m], 6 Sin[ (k-1) Pi/m]},{0,0}}]]
Do[Show[Table[sect[r],{r,1,2m}],pf,tt[[m/2]]],
{m,2,36,2}]
Print["La somme des aires des triangles tend vers  113.097"]
```



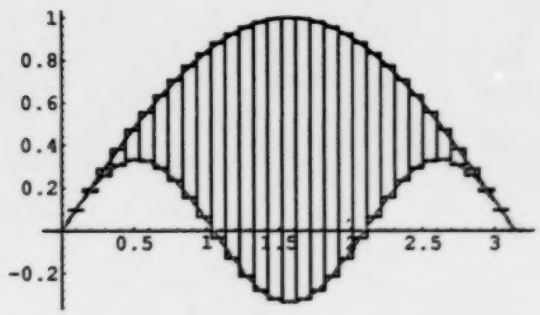

```
La somme des aires des triangles tend vers  113.097
```

Exemple 3.2.10.

Faire un programme qui trace deux courbes de fonctions et qui dessine des rectangles dont la somme des aires calcule l'aire entre les courbes.

Balayage entre deux courbes

```
Clear[sommeRiemann2]

sommeRiemann2[f_,g_,a_,b_,s_]:=Block[{da=(b-a)/s/2,r,ff,gg,k,fs},
  si=2 s;
  ff=Plot[f[x],{x,a,b},PlotStyle->RGBColor[0,0,1]];
  gg=Plot[g[x],{x,a,b},PlotStyle->RGBColor[0,1,0]];
  rec[k_]:=Graphics[Line[{{k-da,g[k]},{k-da,f[k]},{k+da,f[k]},
          {k+da,g[k]},{k-da,g[k]}}]
          ];
Show[Table[rec[r],{r,a,b,2 da}],ff,gg];
fs =2 Sum[ (f[i]- g[i]) da,
      {i,((si-1 )a + b)/si,(a+(si-1 )b)/si,2(b-a)/si}];
  Print["La somme des aires des rectangles est  ",N[fs,5]]
                          ]

f[x_]:=Sin[x];
g[x_]:=1/3 Sin[3 x]


sommeRiemann2[f,g,0,Pi,32]
```




```
La somme des aires des rectangles est  1.7778
```

La vraie valeur:

```
Integrate[Sin[x]-1/3 Sin[3 x],{x,0,Pi}]//N
1.77778
```

□ 3.3 La programmation mathématique.

Cette forme de programmation est aussi appelée programmation récursive. Dans ce genre de programmation on peut définir les objets mathématiques par des règles récursives.

Exemple 3.3.1.

Parmi les exemples classiques de définitions récursives on retrouve la suite de Fibonacci, définie par les règles suivantes:

> Règle 1. Les deux premiers éléments de la suite sont égaux et valent un.
>
> Règle 2. Le nième élément de la suite est la somme des deux précédents.

Avec *Mathematica* la traduction se fait presque littéralement, la correspondance entre les règles de la suite de Fibonacci et la définition de la fonction **fib[ ]** est claire et directe:

```
fib[1]=fib[2]=1;
fib[n_]:=fib[n-1]+fib[n-2]
```

C'est tout. Le langage récursif fait le reste. Il est récursif en se sens qu'il s'appelle lui-même, c'est-à-dire que la fonction **fib[ ]** se retrouve dans la définition même de la fonction **fib[ ]**.

Pour calculer le 11ième terme de la suite on tape:

```
fib[11]
89
```

Remarque.- Tant qu'on n'a pas fait tourner un programme récursif on a toujours l'impression que le programme est incomplet.

Exemple 3.3.2.

La fonction d'Ackermann dont les règles récursives sont:

> Règle 1. A(m, n) = A(m - 1, A(m, n - 1)) pour m et n > 0;
>
> Règle 2. A(0, n) = n + 1;
>
> Règle 3. A(m, 0) = A(m - 1, 1), pour m > 0.

Il suffit d'écrire les mêmes règles dans le langage utilisé par *Mathematica* pour obtenir le programme récursif suivant:

```
ack[m_Integer?Positive,n_Integer?Positive]:=ack[m-1,ack[m,n-1]]
ack[0,n_]:=n+1
ack[m_Integer?Positive,0]:=ack[m-1,1]
```

Il va sans dire que le programme est beaucoup plus rapide, si on l'écrit de cette façon:

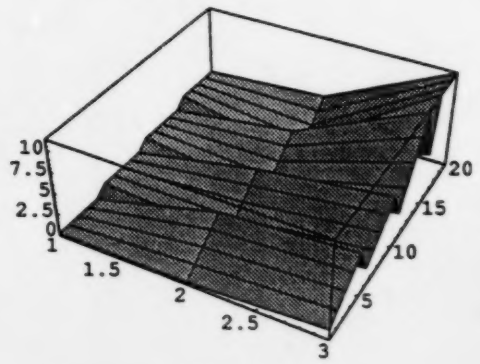
```
ack[m_Integer?Positive,n_Integer?Positive]:=
                                ack[m,n]=ack[m-1,ack[m,n-1]]
ack[0,n_]:=ack[0,n]=n+1

ack[m_Integer?Positive,0]:=ack[m,0]=ack[m-1,1]
```

Voici une *liste* de triplets où l'abscisse varie de 0 à 3, l'ordonnée de 0 à 4 et la cote est **ack[,]**:

```
liste = Partition[Flatten[
              Table[{n,m,ack[n,m]},{n,0,3},{m,0,4}]],
              3]

{{0, 0, 1}, {0, 1, 2}, {0, 2, 3}, {0, 3, 4}, {0, 4, 5}, {1, 0, 2},
  {1, 1, 3}, {1, 2, 4}, {1, 3, 5}, {1, 4, 6}, {2, 0, 3}, {2, 1, 5},
  {2, 2, 7}, {2, 3, 9}, {2, 4, 11}, {3, 0, 5}, {3, 1, 13}, {3, 2, 29},
  {3, 3, 61}, {3, 4, 125}}
```

Mathematica se sert de liste pour produire un graphique en 3D de la fonction d'Ackermann:

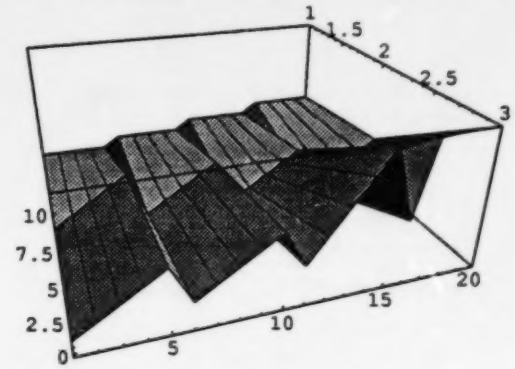
```
ListPlot3D[liste]
```




```
-SurfaceGraphics-
```

Le même graphique sous un autre angle:

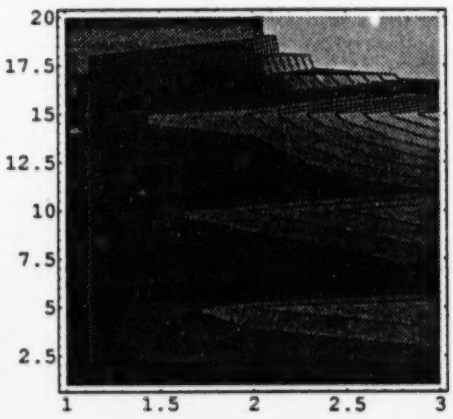
```
Show[%,ViewPoint->{1.5,-0.5,.75}]
```




```
-SurfaceGraphics-
```

Du graphique en 3D *Mathematica* peut tracer une carte des lignes de niveaux:

```
ListContourPlot[liste]
```




```
-ContourGraphics-
```

Et enfin voici la *liste* de cotes de la fonction Ackermann:

```
Flatten[Table[ack[n,m],{n,0,3},{m,0,4}]]
{1, 2, 3, 4, 5, 2, 3, 4, 5, 6, 3, 5, 7, 9, 11, 5, 13, 29, 61, 125}
```

Le calcul de **ack[4, 1]** excède la pile (stack) de 256 octets nécessaires aux appels récursifs. *Mathematica* envoie un message concernant la limitation de ce programme récursif:

```
ack[4,1]
$RecursionLimit::reclim: Recursion depth of 256 exceeded.
$RecursionLimit::reclim: Recursion depth of 256 exceeded.
$RecursionLimit::reclim: Recursion depth of 256 exceeded.
General::stop: Further output of $RecursionLimit::reclim
    will be suppressed during this calculation.
```

On se sert de la fonction d'Ackermann pour étudier la profondeur de la pile ou pour comparer la vitesse du CPU de tel ordinateur à tel autre.

Exemple 3.3.3.

En programmation procédurale nous avions fait un programme qui traduisait les chiffres arabes en chiffres romains. Nous allons faire la même chose, mais cette fois-ci en se basant sur les règles qui définissent les chiffres romains.

> Règle 1. La correspondance entre les lettres et les nombres:
> i=1, v=5, x=10, l=50, c=100, d=500, m=1000, q=5000;
>
> Règle 2. Le chiffre romain de valeur inférieure placé à gauche d'un autre chiffre romain se soustrait, placé à droite il s'additionne.

```
{rom[{i}],rom[{v}],rom[{x}],rom[{l}],
rom[{c}],rom[{d}],rom[{m}],rom[{q}]}=
{1,5,10,50,100,500,1000,5000};

rom[r_]:=If[rom[{r[[1]]}] < rom[{r[[2]]}],
                  -rom[{r[[1]]}]+rom[Drop[r,1]],
                       rom[{r[[1]]}]+rom[Drop[r,1]]
          ];
```

Quel est l'équivalent de MQCCCXXVIII?

De la façon dont le programme est construit la fonction **rom[ ]** ne fonctionne qu'avec des *listes* qu'elle traduit en chiffres arabes par la correspondance établie dans la première partie du programme. On pourrait se servir de la fonction **Input[ ]**, qui prendrait le nombre en chiffres romains tel quel et le changerait en *liste*. Mais comme on ne voulait pas ajouter de lignes supplémentaires afin de conserver l'élégance du programme, voici ce qu'il faut faire pour répondre à la question : «Quel est l'équivalent de MQCCCXXVIII?»

On écrit:

```
z={m,q,c,c,c,x,x,v,i,i,i};
rom[z]

4328
```

Le programme fait-il efficacement les différences? (*le chiffre romain situé à gauche se soustrait...*):

```
z={c,d,x,l,i,v};
rom[z]

444
```

Les sommes?:

```
z={q,m,m,m,c,c,c,l,x,x,x,v,i};
rom[z]

8386
```

Et, finalement, les mixtes?:

```
z={m,c,m,x,c,i,i,i};
rom[z]

1993
```

Commentaires sur les fonctions qui composent le programme:

Drop[,]

Drop[,] est une fonction de *Mathematica* qui agit de la façon suivante:
Drop[*liste*, n] donne *liste* sans ses **n** premiers éléments;
Drop[*liste*, -n] donne *liste* sans ses **n** derniers éléments;
Drop[*liste*, {n}] donne *liste* sans le **n**ième élément;
Drop[*liste*, {m, n}] donne *liste* sans les éléments de **m** à **n**.
Il est clair que **Drop[liste, 1] == Drop[liste,{1}] == Drop[liste, {1, 1}]**.

r[[1]], r[[2]]

r[[n]] ou **Part[r, n]** sont équivalents, ils donnent le **n**ième élément de la *liste* **r**. Dans le programme, la fonction **If[ ]** compare les deux premiers éléments de la *liste* **r**, qui représentent le chiffre romain. Si le premier élément est plus petit que le deuxiième élément la fonction **If[ ]** soustrait la valeur du premier élément du deuxième élément et enlève le premier élément de la *liste* **r**, sinon, elle les additionnne et ensuite enlève le premier élément. La fonction **rom [{ }]** étant récursive le même procédé s'applique au deuxième et au troisième élément, et ainsi de suite pour tous les éléments de la liste **r**.

Exemple 3.3.4.

Les tours de Hanoï.

Selon une légende, les moines bouddhistes y jouent avec des pieux et des disques, et la fin de la partie symbolise et annonce la fin du monde. Ils disposent de trois tours où on y empile des disques de pierre de diamètres décroissants. Il y en a en tout 64. Le but du jeu est de faire passer les disques de la tour n° 1 à la tour n° 3, en ne déplaçant qu'un seul disque à la fois, et en utilisant la tour intermédiaire n° 2 de telle sorte qu'à aucun moment un disque ne soit empilé sur un disque plus petit.

```
m="Le disque numéroté ";
v=" va de ";
r=" à ";
tr[n_,to_,ta_,ti_]:= If[n > 0,
                        tr[n-1,to,ti,ta];
                        Print[m,to,v,to,r,ta];
                        tr[n-1,ti,ta,to]]
```

Dans **tr[n,to,ta,ti]** les variables se traduisent par:

tr: la fonction tour;
n: le nombre de disques;
to: la tour d'origine;
ta: la tour d'arrivée;
ti: la tour intermédiaire.

Faisons un essai du jeu avec **trois** disques, où la tour *un* est la tour origine, la tour *trois* la tour arrivée et la tour *deux* la tour intermédiaire.

```
tr[3,1,3,2]

Le disque numéroté 1 va de 1 à 3
Le disque numéroté 1 va de 1 à 2
Le disque numéroté 3 va de 3 à 2
Le disque numéroté 1 va de 1 à 3
Le disque numéroté 2 va de 2 à 1
Le disque numéroté 2 va de 2 à 3
Le disque numéroté 1 va de 1 à 3
```

De la façon dont le programme est fait, n disques produiront 2^n - 1 déplacements (64 disques, 2^{64} - 1 déplacements):

```
2^64 - 1//N

1.84467 10^19
```

Pour se faire une représentation mentale de ce nombre impressionnant, faisons le calcul suivant. Si on suppose qu'il est possible de faire, en moyenne, un déplacement par seconde, 24 heures par jour pendant cent ans, la division suivante nous donne le nombre de siècles qu'il y a dans $2^{64} - 1$ secondes:

```
(2^64-1)/60/60/24/365/100//N
```

$5.84942\ 10^{9}$

Près de six milliards de siècles.

Exemple 3.3.5.

En programmation fonctionnelle nous avions fait un programme qui nous donnait toutes les composantes des hypercubes unités. Après avoir analysé la matrice des résultats on remarque que le 80 de la ligne {5, D} est obtenu par le 24, de la ligne {4, D}, multipié par 2 auquel on ajoute le 32 qui est situé à gauche du 24 . On y voit le schéma suivant:

$$32 + 24 * 2$$
$$80$$

Si on se déplace dans la matrice tous les nombres du tableau se comportent selon ce schéma. Le 84 de la ligne {7, D] est égale à 60 + 12*2, ... ce qui nous a amenés à découvrir les règles récursives de ce programme:

Règle 1. hyper(0, 0) = 1;

Règle 2. hyper(0, k) = 1, si k > 0;

Règle 3. hyper(n , 0) = 2 hyper(n-1, 0), si n > 0 ou n = 0;

Règle 4. hyper(n, k) = hyper(n-1, k-1) +2 hyper(n-1, k), si n > 0 et k > 0.

La traduction dans le langage compris par *Mathematica*:

```
hyper[0,0]=1;
hyper[0,k_Integer?Positive]:=0;
hyper[n_Integer?NonNegative,0]=2 hyper[n-1,0];
hyper[n_Integer?Positive,k_Integer?Positive]:=
                    hyper[n-1,k-1] + 2 hyper[n-1,k];
```

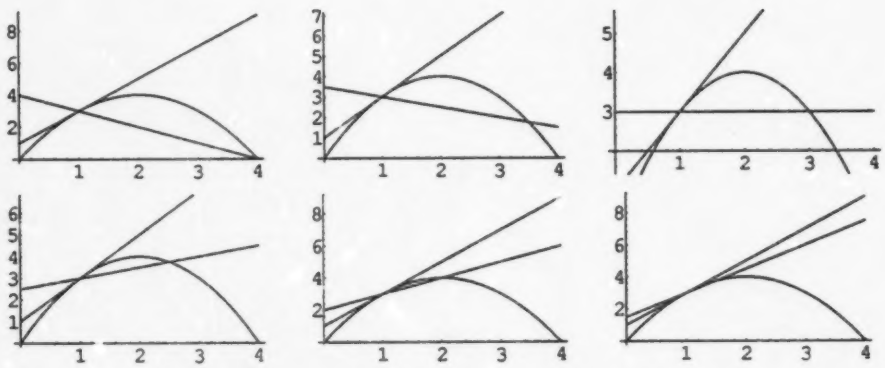
Et pour plus de rapidité, écrivons:

```
hyper[0,0]=1;
hyper[0,k_Integer?Positive]:=hyper[0,k]=0;
hyper[n_Integer?NonNegative,0]:=hyper[n,0]=2 hyper[n-1,0];
hyper[n_Integer?Positive,k_Integer?Positive]:=hyper[n,k]=
                    hyper[n-1,k-1] + 2 hyper[n-1,k];

MatrixForm[Table[hyper[n,k],{n,0,9},{k,0,n}]]
```

```
{1}
{2, 1}
{4, 4, 1}
{8, 12, 6, 1}
{16, 32, 24, 8, 1}
{32, 80, 80, 40, 10, 1}
{64, 192, 240, 160, 60, 12, 1}
{128, 448, 672, 560, 280, 84, 14, 1}
{256, 1024, 1792, 1792, 1120, 448, 112, 16, 1}
{512, 2304, 4608, 5376, 4032, 2016, 672, 144, 18, 1}
```

Quatrième partie

Les projets - Calcul I



▫ Chapitre I.

1.1 La définition de fonction

Une variable y est une fonction de la variable x, si une relation existe entre eux telle que pour une valeur donnée de x correspond une et une seule valeur de y.

Exemple 1.1.1.

Un réservoir de 64 litres est rempli avec une solution de 20% d'alcool. Supposons que nous vidangeons x litres de la solution et que nous la remplaçons par de l'alcool pur. Exprimer la nouvelle concentration de la solution comme une fonction de x. À partir du graphe de la concentration discuter la façon dont elle varie avec x.

Solution:

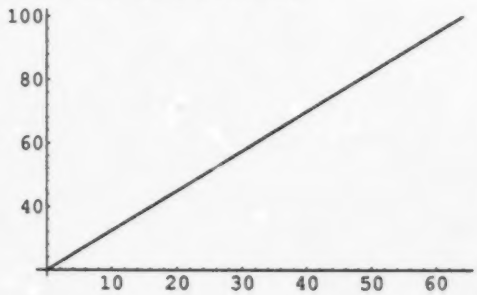
Après avoir enlevé x litres de la solution, il reste **(64 - x)** litres à **20%**, i.e. **(64 - x)/5** litres d'alcool pur.

On ajoute x litres d'alcool pur, ce qui nous donne un total de **[(64 -x)/5 + x]** d'alcool pur.
La nouvelle concentration est donc:

$$c[x_] := \frac{(\frac{64 - x}{5} + x)\ 100}{64}$$

La fonction est une droite que nous pouvons tracer avec la fonction **Plot[ ]**:

```
Plot[c[x], {x, 0, 64}]
```




```
-Graphics-
```

La simplification de **c[x]** donne **20 + 5/4 x**, qui est une fonction affine.

Exemple 1.1.2.

Soit n un entier > 1, posons:

f(n) = le plus grand facteur premier de n.

Un nombre premier est tout entier différent de un qui ne possède aucun diviseur, sauf lui-même et un.

Peut-on dire que cette définition correspond aux exigences demandées par la définition d'une fonction? Peut-on l'exprimer analytiquement? Construire le graphe pour n = 2 jusqu'à n = 50.

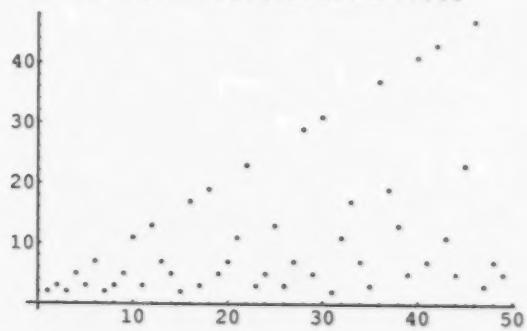
Solution:

A chaque entier plus grand que un correspond une et une seule image:

```
f[n_]:=Max[Table[FactorInteger[n][[i]][[1]],
                 {i,1,Length[FactorInteger[n]]}]];
Table[{v,f[v]},{v,2,50}]
```

```
{{2, 2}, {3, 3}, {4, 2}, {5, 5}, {6, 3}, {7, 7}, {8, 2}, {9, 3},
  {10, 5}, {11, 11}, {12, 3}, {13, 13}, {14, 7}, {15, 5}, {16, 2},
  {17, 17}, {18, 3}, {19, 19}, {20, 5}, {21, 7}, {22, 11}, {23, 23},
  {24, 3}, {25, 5}, {26, 13}, {27, 3}, {28, 7}, {29, 29}, {30, 5},
  {31, 31}, {32, 2}, {33, 11}, {34, 17}, {35, 7}, {36, 3}, {37, 37},
  {38, 19}, {39, 13}, {40, 5}, {41, 41}, {42, 7}, {43, 43}, {44, 11},
  {45, 5}, {46, 23}, {47, 47}, {48, 3}, {49, 7}, {50, 5}}
```

```
ListPlot[Table[f[w],{w,2,50}]]
```




```
-Graphics-
```

Exemple 1.1.3.

Soit n un entier positif, posons

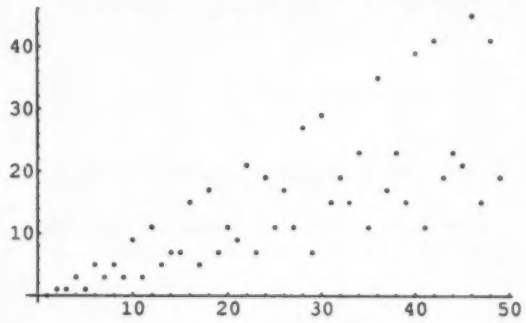
f(n) = le nombre d'entiers positifs qui ne sont pas plus grand que n et qui sont premiers à n.

Construire le graphe de f(n) de n = 2 à n = 50.

```
facPrime[n_]:=Table[FactorInteger[n][[i]][[1]],
                    {i,1,Length[FactorInteger[n]]}]
cf[n_]:=Count[Table[Intersection[facPrime[k],facPrime[n]],
                    {k,2,n}],{}]
Table[{w,cf[w]},{w,1,50}]
```

```
{{1, 0}, {2, 0}, {3, 1}, {4, 1}, {5, 3}, {6, 1}, {7, 5},
  {8, 3}, {9, 5}, {10, 3}, {11, 9}, {12, 3}, {13, 11},
  {14, 5}, {15, 7}, {16, 7}, {17, 15}, {18, 5}, {19, 17},
  {20, 7}, {21, 11}, {22, 9}, {23, 21}, {24, 7}, {25, 19},
  {26, 11}, {27, 17}, {28, 11}, {29, 27}, {30, 7},
  {31, 29}, {32, 15}, {33, 19}, {34, 15}, {35, 23},
  {36, 11}, {37, 35}, {38, 17}, {39, 23}, {40, 15},
  {41, 39}, {42, 11}, {43, 41}, {44, 19}, {45, 23},
  {46, 21}, {47, 45}, {48, 15}, {49, 41}, {50, 19}}
```

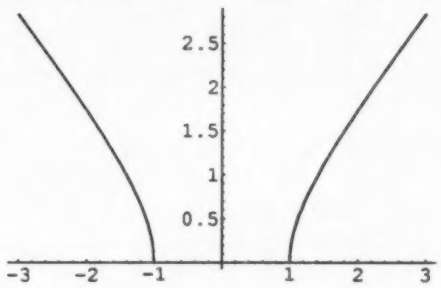
```
ListPlot[Table[cf[w],{w,2,50}]]
```



1.2 Le domaine de définition.

On appelle domaine de définition d'une fonction réelle toutes les valeurs de x telles que f(x) existe en tant que nombre réel. Le domaine de définition de f(x) = Sqrt[x^2 - 1] est le sous-ensemble des réels tel que x^2 - 1 est supérieur ou égal à zéro:

```
Plot[Sqrt[x^2-1],{x,-3,3}]
```




```
-Graphics-
```

Pour tracer le graphe d'une fonction, les valeurs de la variable doivent être prises dans le domaine de la fonction. Si on regarde le graphique, *Mathematica* exclut toutes les valeurs entre -1 et 1. Le domaine de définition de la fonction est l'ensemble des réels sauf les nombres entre -1 et 1. Si nous n'avions pas eu le graphe de la fonction comment aurions-nous pu déterminer le domaine de la fonction?

Voici les règles qu'il faut connaître pour déterminer le domaine d'une fonction:

> Règle 1. Si la fonction s'écrit sous forme d'un quotient, le dénominateur doit être différent de zéro.
>
> Règle 2. Si la fonction s'écrit avec un radical, la quantité sous le radical ne doit pas être négative.
>
> Règle 3. Si la fonction est une fonction logarithmique, l'argument doit être positif.
>
> Règle 4. Si la fonction est la composée de deux autres fonctions le domaine de définition est l'image réciproque de l'intersection de l'image de cette fonction avec le domaine de l'autre fonction ou vice versa.

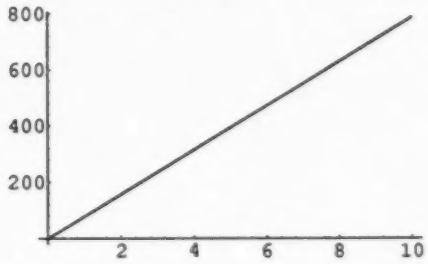
1.3 La croissance d'une fonction.

Exemple 1.3.1.

Graphiquement, une fonction croît si son graphe monte et décroît si son graphe descend.

Soit la fonction qui donne le volume d'une boîte cylindrique de rayon cinq en fonction de la hauteur, son volume est défini par $25\,\pi\,h$:

```
v[h_]:= 25 Pi h

Plot[v[h],{h,0,10}]
```



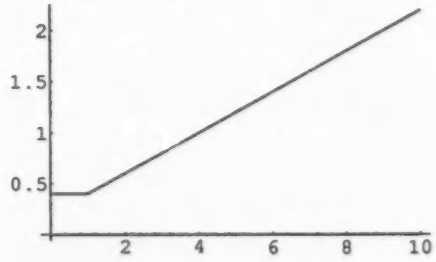
Cette fonction est une fonction croissante, elle est directement proportionnelle à h, i.e. plus la hauteur augmente, plus le volume augmente.

Exemple 1.3.2.

Soit la fonction coût d'un envoi postal définie par:

toute enveloppe qui pèse moins d'une once: 0.40 $ (c = .40; $0 < p < 1$)
celle qui pèse une once ou plus: 0.20 + 0.20(once) $ (c = .20 + .20 p; $p > 1$ ou $p = 1$)

```
c[p_Real?Positive]:=If[p>=1,.20+.20 p,.40]

Plot[{c[x],0},{x,0,10}]
```




```
-Graphics-
```

Cette fonction est une fonction croissante, plus exactement nous dirons une fonction non décroissante puisqu'elle croît tout en ayant une partie constante.

1.4 L'accroissement

Exemple 1.4.1.

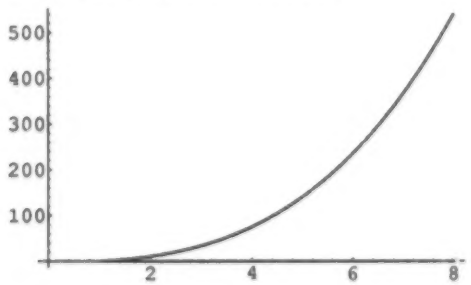
Quand l'argument d'une fonction passe de x = 1 à x = 8, on dit qu'il a subi un accroissement de sept unités. Si la fonction est $f(x) = x^3 + 4x - 5$, par exemple, elle aussi va subir un accroissement qui correspond à la différence des images aux points x = 8 et x = 1:

```
Clear[f];

f[x_]:= x^3 + 4 x - 5;

f[8]-f[1]
539
```

Donc, la fonction prend un accroissement $\Delta y = f(8) - f(1)$ égal à 539.

Si une fonction est croissante, alors la fonction a un accroissement positif sur l'intervalle qui lui correspond.

```
Plot[{f[x],0},{x,1,8}]
```




```
-Graphics-
```

Quand une variable passe de x_1 à x_2, l'accroissement $(x_2 - x_1)$ est symbolisé par $\Delta x = x_2 - x_1$.
L'accroissement de la fonction est symbolisé par $\Delta y = f(x_2) - f(x_1)$.

Une autre façon d'exprimer la même idée:
Si une variable x prend un accroissement Δx au point x = a, la fonction prend l'accroissement $\Delta y = f(a + \Delta x) - f(a)$.

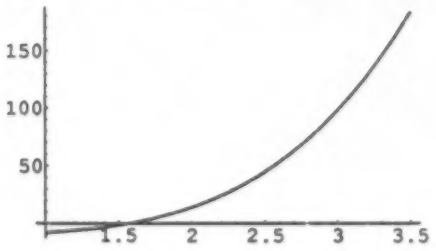
Exemple 1.4.2.

Soit la fonction $f(x) = x^4 + x^3 - 10$, si la variable prend un accroissement $\Delta x = 2$ au point $x = 1$, quel est l'accroissement de la fonction?

```
Clear[f];
a=1;
delta=2;
f[x_]:= x^4 + x^3 - 10;
f[a + delta]- f[a] (* l'accroissement de la fonction *)
```

106

```
Plot[{0,f[x]},{x,1,3.5}]
```



-Graphics-

```
f[3]-f[1]
```

106

1.5 Le maximum et le minimum

On dit qu'une fonction a une valeur maximale au point $x = a$, si $f(x) < f(a)$ pour tous les x dans le voisinage de a.

On dit qu'une fonction a une valeur minimale au point $x = a$, si $f(x) > f(a)$ pour tous les x dans le voisinage de a.

Exemple 1.5.1.

Soit $f(x) = 1 - x^2$, cette fonction a une valeur maximale au point $x = 0$ qui vaut 1:

```
Clear[f];

f[x_]:= 1 - x^2;

m=Table[f[x],{x,-1,1,.125/2}]

{0, 0.121094, 0.234375, 0.339844, 0.4375, 0.527344, 0.609375,
  0.683594, 0.75, 0.808594, 0.859375, 0.902344, 0.9375, 0.964844,
  0.984375, 0.996094, 1., 0.996094, 0.984375, 0.964844, 0.9375,
  0.902344, 0.859375, 0.808594, 0.75, 0.683594, 0.609375, 0.527344,
  0.4375, 0.339844, 0.234375, 0.121094, 0.}

Max[m]

1.
```

Exemple 1.5.2.

Soit la fonction d(x) = Sqrt[2] / 2 (x^2 + 345 / x)

$$\frac{\text{Sqrt[2]}\ \left(x^2 + \frac{345}{x}\right)}{2}$$

Quelle est sa valeur minimale?

```
d[x_]:=Sqrt[2]/2(x^2 + 345/x);

m=Table[N[d[x]],{x,5,7,1/100}];

Min[m]

65.7354
```

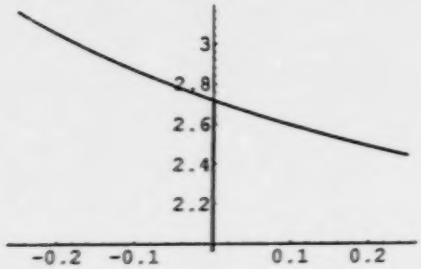
La valeur minimale est 65.7354, c'est le minimum de la fonction. La valeur de x qui donne cette valeur minimale est 5.566681407604759137 ou (345/2)^(1/3) Nous verrons dans les chapitres à venir quelle est la méthode utilisée pour trouver précisément la valeur de x qui donne le minimum.

Chapitre 2.

2.1 La limite d'une fonction.

Il existe des fonctions qui n'ont pas d'image en un point précis. Considérons la fonction (1+1/x)^x, cette fonction n'a pas d'image au point x = 0. Elle n'est pas définie à zéro, mais elle l'est pour tous les points autour de zéro. Donc, si on s'approche très près de zéro, quel sera le comportement des images dans ce voisinage? Ces nombres produiront-ils des images qui s'approchent elles aussi d'une valeur précise? ou produiront-ils des images qui deviendront de plus en plus grandes?

```
Clear[f];

f[x_]:=(1+x)^(1/x);

Plot[f[x]//N,{x,-.25,.25}]
```




```
-Graphics-
```

Dans le voisinage de zéro la fonction est continue, sauf à zéro. Nous allons faire un essai avec un nombre très petit près de 0, 10^{-9}, par exemple:

```
N[f[.0000000001],10]

2.718281829
```

Donc on voit que lorsque x est très près de 0, f[x] s'approche de 2.718281829... En mathématique nous exprimons ce fait en disant que la limite de f de x quand x s'approche de 0 est 2.718281828. On écrit:

$$\lim_{x \to 0} f(x) = e = 2.718281828...$$

Effectuons la vérification avec la fonction **Limit[ ]**:

```
Limit[(1+x)^(1/x),x->0]

E
```

2.2 Le nombre e.

La fonction S[ ] définie plus bas est la somme des inverses des factorielles, elle possède une limite que nous ne pouvons déterminer à l'avance.

La fonction G[ ] est la série géométrique de raison 1/2 qui converge vers quatre.

La fonction T[ ] est la fonction dont nous avons étudié la limite plus haut.

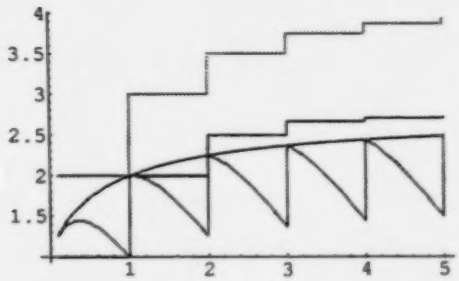
La fonction B[ ] est la fonction T[ ] développée selon le binôme de Newton.

Nous allons étudier graphiquement ces quatre fonctions. La fonction géométrique est la plus grande de toutes ces fonctions et devient en quelque sorte une borne supérieure pour les trois autres. Elle les majore toutes; nous pouvons affirmer que les trois autres fonctions convergent quand leur argument tend vers l'infini.

```
S[n_]:=Sum[1/i!,{i,0,n}]

G[n_]:=Sum[1/2^(i-1),{i,0,n}]

T[n_]:=(1 + 1/n)^n

B[n_]:=Sum[Binomial[n,r](1/n)^(n-r),{r,0,n}]
```

Ce graphique prend toutes les valeurs réelles. La fonction B[ ] devrait être définie sur des entiers.

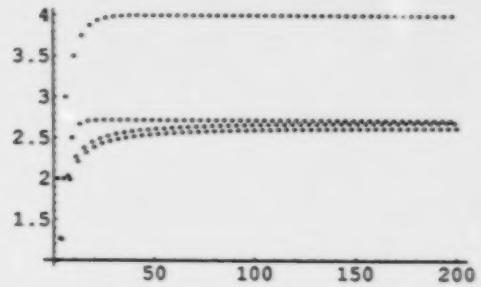
```
Plot[{S[n],G[n],T[n],B[n]},{n,.1,5},
PlotStyle->{RGBColor[0,0,1],RGBColor[0,1,0],
            RGBColor[1,0,0],RGBColor[1,0,1]}]
```



-Graphics-

Dans le graphique qui suit nous ne considérons que les valeurs prises sur les entiers:

```
fonc=Table[{S[n],G[n],T[n],B[n]}//N,{n,.1,50}];
ListPlot[Flatten[fonc]]
```




```
-Graphics-
```

La convergence est évidente (du moins visuelle). Les fonctions **S**, **T** et **B** tendent vers une limite unique. Cette limite s'appelle **e**, la base des logarithmes népériens:

```
N[E,50]
2.71828182845904523536028747135266249775724709 37
```

Lorsqu'on doit démontrer de façon rigoureuse cette convergence on se sert des fonctions définies plus haut: **S**, **G**, **T** et **B**.

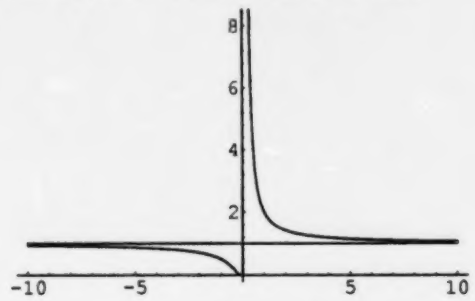
2.3 La limite à l'infini.

Quand un nombre augmente indéfiniment nous disons qu'il tend vers l'infini:

```
Limit[2^(1/x),x->Infinity]
1
```

Le calcul précédent nous montre que la courbe s'approche d'une valeur constante quand x tend vers l'infini. On exprime ce fait en disant que le graphe possède une asymptote horizontale.

```
Plot[{1,2^(1/x)},{x,-10,10}]
```



2.4 La définition de limite.

La définition d'une limite s'énonce comme suit:

$\lim_{x \to a} f(x) = L \Leftrightarrow \forall \varepsilon > 0, \exists \delta > 0$ tel que $0 < |x - a| < \delta \Rightarrow |f(x) - L| < \varepsilon$

Les nombres δ et ε sont de très petits nombres positifs.

$|x - a| < \delta$ se traduit par *x est très près de a aussi bien à gauche qu'à droite*;

$0 < |x - a| < \delta$ se traduit par *non seulement x est près de a, mais il est différent de a*;

$|f(x) - L| < \varepsilon$ se traduit par *f(x) est très près de L*;

$\forall \varepsilon > 0, \exists \delta > 0$ se traduit par *peu importe la valeur de* ε, *on peut toujours trouver un* δ.

L'ensemble de la définition donne:

La limite d'une fonction f(x) est L, si et seulement si, pour tout ε positif, il existe un δ positif tel que x près de a ($x \neq a$) implique que f(x) est près de L.

C'est une définition conceptuelle, non fonctionnelle ou opérationnelle. On peut difficilement se servir de cette définition pour trouver techniquement une limite. On s'en sert surtout pour la démonstration des théorèmes sur les limites.

2.5 La définition de continuité.

Une fonction est continue au point x = a, si la limite de la fonction quand x tend vers a est égale à l'image de la fonction au point x = a:

$$\lim_{x \to a} f(x) = f(a)$$

Exemple 2.5.1.

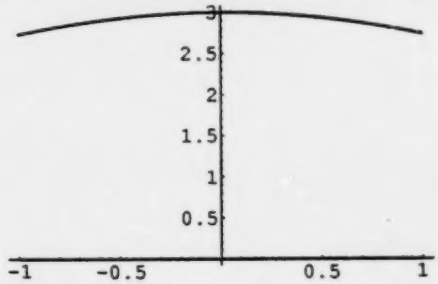
La fonction f(x) = (4x + 5 sin(x)) / 3x n'est pas définie au point x = 0. Quelle image doit-on lui donner si nous voulons qu'elle soit continue?

```
Limit[(4x + 5 Sin[x])/(3 x),x->0]
```

```
3
```

Il faut définir f(0) = 3 pour avoir une fonction continue.

```
Plot[{0,(4x + 5 Sin[x])/(3 x)},{x,-1,1}]
```




```
-Graphics-
```

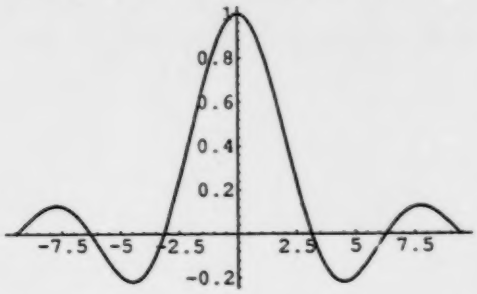
Exemple 2.5.2.

La fonction f(x) = sin(x) / x, n'est pas définie au point x = 0. Peut-elle être continue en lui donnant une valeur appropriée à x = 0?

```
Limit[Sin[x]/x,x->0]
```

```
1
```

```
Plot[Sin[x]/x, {x, -3Pi, 3Pi}]
```

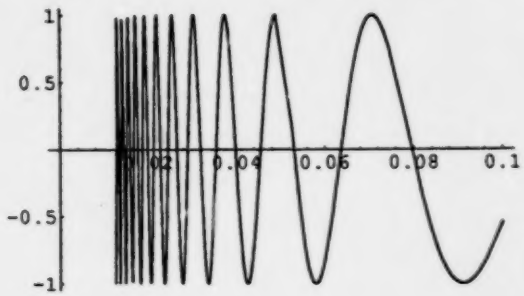



```
-Graphics-
```

Exemple 2.5.3.

Nous allons faire l'étude du graphe de f(x) = sin(1 / x).

```
Plot[Sin[1/x], {x, .0125, .1}]
```



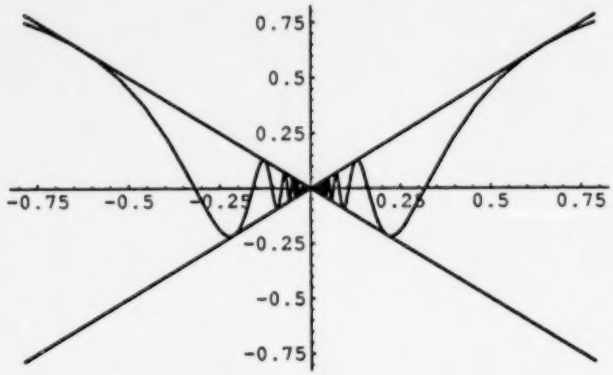

```
-Graphics-
```

Cette fonction prend toutes les valeurs entre -1 et +1 quand le nombre 1/x se situe entre $(2n - 1/2)\pi$ et $(2n + 1/2)\pi$, peu importe la valeur de n. Quand $x = 2/(4n-1)\pi$ la fonction prend la valeur -1 et quand $x = 2/(4n+1)\pi$ la fonction prend la valeur +1. Dans le voisinage immédiat de zéro la fonction oscille entre -1 et +1 très rapidement comme on peut le voir dans le graphe ci-dessus.

Exemple 2.5.4.

Traçons le graphe de f(x) = x sin(1 / x). Nous allons montrer que la fonction est continue à 0, si nous posons f(0) = 0.

```
Plot[{-x,x Sin[1/x],x},{x,-Pi/4,Pi/4}]
```




```
-Graphics-
```

```
Limit[x Sin[1/x],x->0]
```

```
0
```

Exemple 2.5.5.

Dans ce qui suit on trouvera des fonctions et des graphiques pour visualiser les limites dans les calculs techniques.

```
f[x_]:= x^3/(x^2-1)
```

```
{f[.9],f[.99],f[.999],f[.9999],f[.99999],f[.999999],
f[.9999999],f[.99999999],f[.999999999],f[.9999999999],
f[.99999999999]}
```

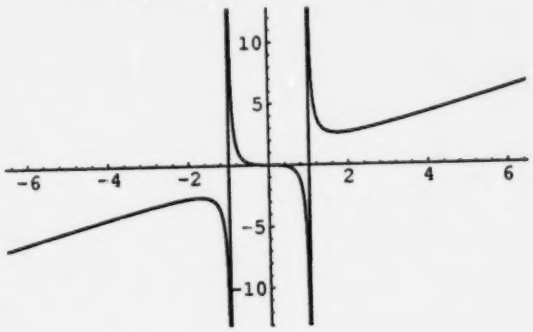
```
{-3.83684, -48.7587, -498.751, -4998.75, -49998.8,
  -499999., -5. 10^6, -5. 10^7, -5. 10^8, -5. 10^9, -5. 10^10}
```

D'où la limite à gauche de 1 est moins l'infini:

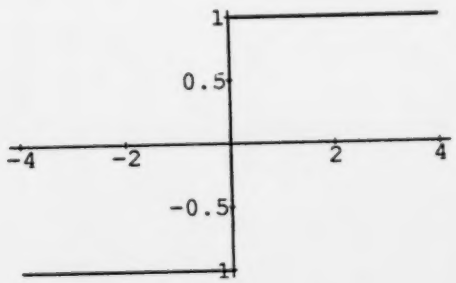
```
Limit[f[x],x->1,Direction->1]
```

```
-Infinity
```

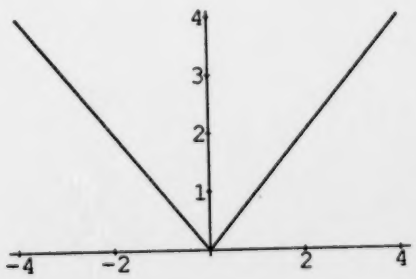
```
Plot[f[x],{x,-10,10}]
```




```
-Graphics-
g[x_]:=If[x>0,1,-1]

Plot[g[x],{x,-4,4}]
```

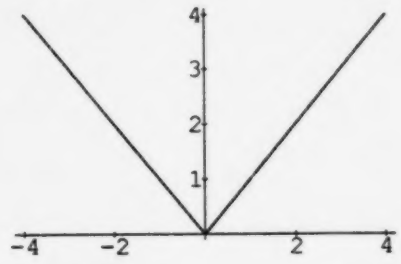



```
-Graphics-
h[x_]:=If[x>=0,x,-x]

Plot[h[x],{x,-4,4}]
```



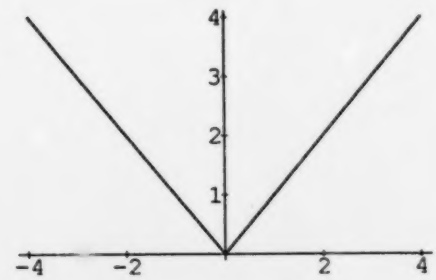

```
-Graphics-
```

```
Plot[Abs[x],{x,-4,4}]
```



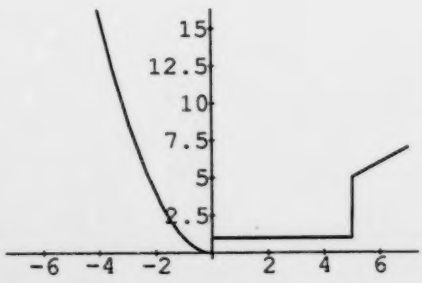
-Graphics-

```
va[x_]:=x/;x>=0;va[x_]:=-x/;x<0

Plot[va[x],{x,-4,4}]
```



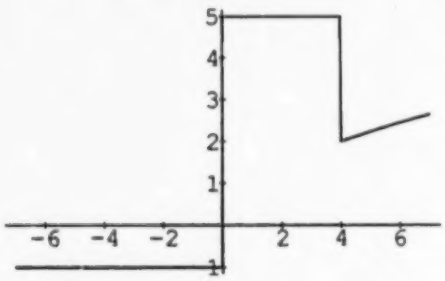
-Graphics-

```
fb[x_]:=Which[x<0,x^2,x>5,x,True,1]

Plot[fb[x],{x,-7,7}]
```



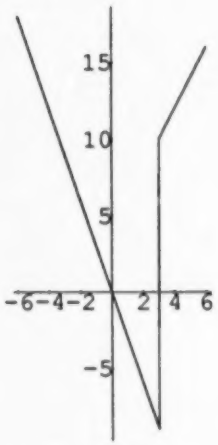
-Graphics-

```
fsp[x_]:=Which[x<0,-1,x>4,Sqrt[x],True,5]

Plot[fsp[x],{x,-7,7}]
```



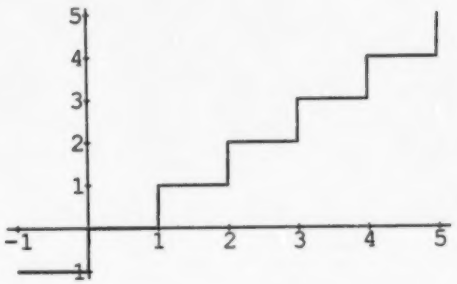
-Graphics-

```
Clear[fdg];
fdg[x_]:=2x+4/;x>3
fdg[x_]:=+2/;x=3
fdg[x_]:=-3x/;x<3

Plot[fdg[x],{x,-6,6},AspectRatio->Automatic]
```



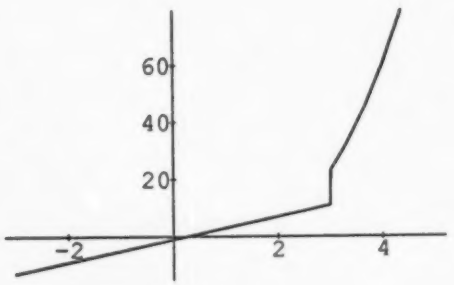
-Graphics-

```
Plot[Floor[x],{x,-1,5}]
```




```
-Graphics-
```

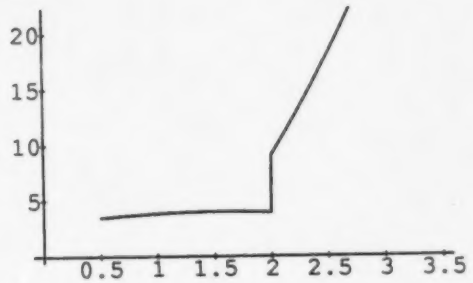
```
fbsp[x_]:=Which[x>=3,x^3-4,True,4x-1]

Plot[fbsp[x],{x,-3,5}]
```




```
-Graphics-
```

```
ff[x_]:=3x^2+5/;x>=5
ff[x_]:=4x^2-7/;x>2 && x<5;
ff[x_]:=3+Sin[x]/;x>0 && x<=2;
ff[x_]:=-3/;x<0;

ff[2.000000000001]

9.

ff[2]//N

3.9093

Limit[ff[x],x->1.99999999]

3.9093
```

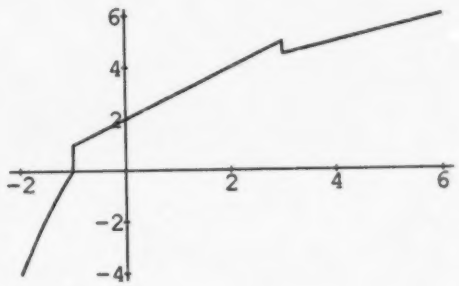
```
Plot[ff[x],{x,.5,3.5}]
```




```
-Graphics-
```

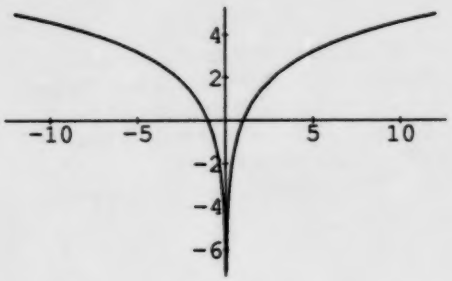
```
Clear[fgf];
fgf[x_]:=(1/2)x+3 /;x>=3;
fgf[x_]:=x+2 /;x>=-1&&x<3;
fgf[x_]:=-x^2+x+2 /;x<-1;
```

```
Plot[fgf[x],{x,-2,6}]
```




```
-Graphics-
```

```
Plot[Log[x^2],{x,-12,12}]
```




```
-Graphics-
```

```
Limit[Log[x^2],x->0]
```

```
-Infinity
```

Remarque.- Les graphes que nous venons de voir n'ont d'intérêt que si nous voulons étudier les limites spéciales de fonctions particulières.

▫ Chapitre 3.

3.1 La dérivée.

L'accroissement d'une fonction divisé par l'accroissement de la variable qui tend vers zéro s'appelle la dérivée de la fonction. En d'autres mots, la dérivée est le taux de variation instantané d'une fonction. Le concept de dérivée, comme celui de l'intégrale, est d'origine intuitive. Le problème qui consiste à construire une tangente à une courbe donnée et celui de trouver une définition précise de la notion de vitesse dans un mouvement donné sont les pierres d'achoppement qui ont conduit à l'idée de dérivée. La théorie différentielle a contribué à aplanir ces difficultés.

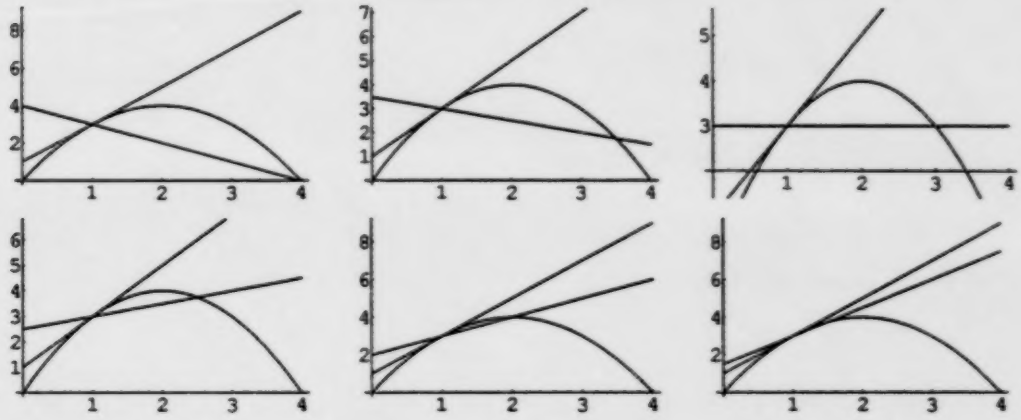
3.2 La dérivée et la tangente.

On appelle sécante à une courbe une droite qui coupe cette courbe en un ou plusieurs points. Pour arriver à conceptualiser l'idée de tangente imaginons que les points sécants s'approchent d'un des points. Il va venir un temps où les points seront tellement proches les uns des autres que la sécante va se confondre avec la tangente. D'où la tangente en un point est la droite qui est la position limite d'une sécante. Le point s'appelle le point de tangence.

Dans ce qui suit nous allons faire quelques graphiques qui illustrent l'idée de sécantes qui se métamorphosent en tangente. La sécante tourne autour du point de rencontre de la courbe avec la droite pour finalement devenir une tangente.

```
f[x_]:=x(4 - x);

d[x_]:=f[1] + f'[1](x-1);

s[x_]:=f[1] + m(x-1);


graphes=Table[Plot[{f[x],d[x],s[x]},{x,0,4}]
,{m,-1,3/2,1/2}];
```

```
Show[GraphicsArray[{{graphes[[1]],graphes[[2]],graphes[[3]]},
{graphes[[4]],graphes[[5]],graphes[[6]]}}]]
```



```
-GraphicsArray-
```

La pente de la sécante qui passe par (x, f(x)) et (1, f(1)) est:

$$\frac{f[x] - f[1]}{x - 1}$$

Définition:

La pente de la tangente au point (1, f(1)) est:

```
Limit[(f[x] - f[1])/(x - 1), x -> 1]
```

Exemple 3.2.1.

Si on reprend l'exemple plus haut:

```
f[x ]:=x(4-x)
```

```
Limit[(f[x] - f[1])/(x - 1), x -> 1]
```

```
2
```

La pente de la tangente au point x = 1 est 2.

Nous pouvons aussi définir la dérivée par:

```
Limit[(f[h + 1] - f[1])/h, h -> 0]
```

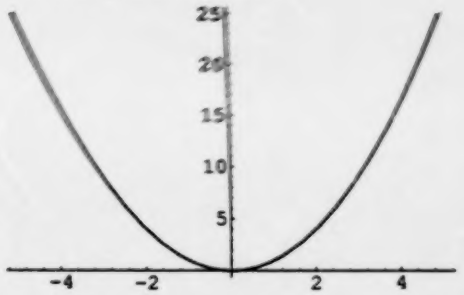
Il suffit de poser h = x - 1 dans la définition.

Exemple 3.2.2.

Traçons le graphe de $f(x) = x^2$:

```
f[x_] := x^2
```

```
Plot[f[x], {x, -5, 5}]
```

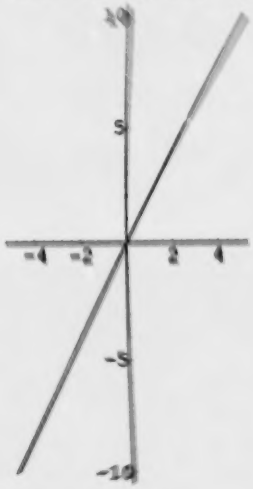



```
-Graphics-
```

Maintenant, traçons le graphe de g(x) = (f(x+h)-f(x))/h, où h = .001

```
g[x_] := (f[x + h] - f[x])/h
```

```
h=.001;
Plot[g[x], {x, -5, 5}, AspectRatio->Automatic]
```




```
-Graphics-
```

C'est le graphe de la fonction dérivée. La fonction g est la limite de (f(x + h) - f(x))/h quand h tend vers zéro.

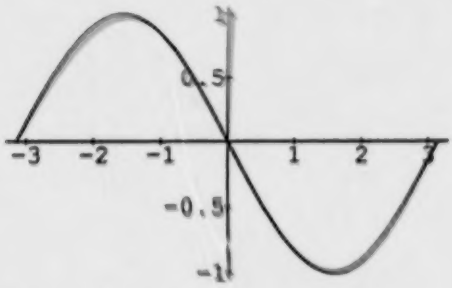
Exemple 3.2.3.

Le graphe de la dérivée et celui du taux de variation de la fonction se confondent, si l'accroissement de la variable est relativement petit.

Voici un exemple avec l'opposé de la fonction sinus et le taux de variation de la fonction cosinus.

Le taux de variation de la fonction cosinus où la variation de la variable est h = 0.001.

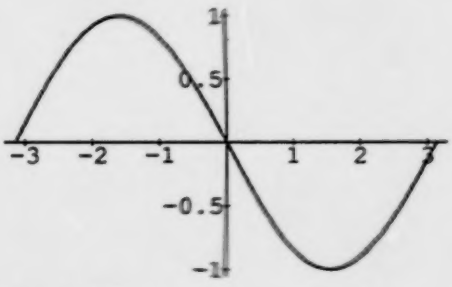
```
h=.001
dc=Plot[(Cos[x+h]-Cos[x])/h,{x,-Pi,Pi},
                    PlotStyle->RGBColor[0,0,1]]
```




```
-Graphics-
```

Le graphe de la fonction y = - sin x:

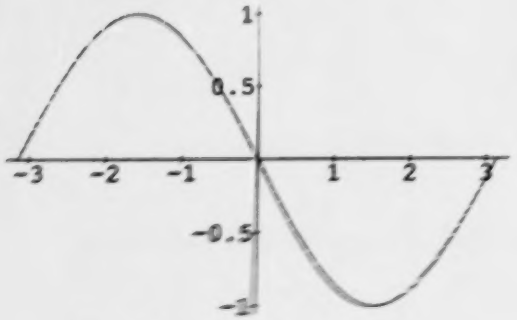
```
gs=Plot[-Sin[x],{x,-Pi,Pi},
                    PlotStyle->RGBColor[1,0,0]]
```




```
-Graphics-
```

Les deux graphes sont superposés dans un même graphique. Comme ils sont de couleurs différentes, l'écran couleur nous laisse voir des différences à peine perceptibles. Aussi, dirons-nous qu'ils se confondent, si la variation de la variable est petite.

```
Show[dc, gs]
```



-Graphics-

3.3. La dérivée et la vitesse.

Exemple 3.3.1.

La vitesse moyenne est le quotient du déplacement par le temps pris pour le parcourrir. Si cette vitesse moyenne tend vers une limite finie quand le temps pris pour le parcourir est de plus en plus petit, cette limite est la vitesse au temps t.

En d'autres mots, la vitesse au temps t est la dérivée suivante:

```
f'[t] = Limit[(f[u] - f[t])/(u - t), u -> t]
```

Ou bien

```
f'[t] = Limit[(f[t + h] - f[t])/h, h -> 0]
```

Pour faire le lien entre mouvement et vitesse considérons le cas d'un corps qui tombe en chute libre. L'expérimentation nous montre qu'un corps qui tombe en chute libre durant un temps t a un déplacement proportionnel à t^2. Ce phénomène peut se représenter par une fonction de la forme:

```
f[t_] := alpha t^2
```

où alpha est une constante de proportionnalité.

La vitesse qui est la dérivée de cette fonction est:

```
f'[t]
2 alpha t
```

Ceci montre que la vitesse d'un corps en chute libre est proportionnelle au temps t. Cette constatation est aussi vérifiée par l'expérimentation.

Mathematica dérive toutes fonctions continues et lisses. Voici une suite d'exemples permettant de mettre le logiciel sur la sellette:

Exemple 3.3.2.

```
D[(x+1)^4,x]
4 (1 + x)^3
D[Sin[x^5],x]
5 x^4 Cos[x^5]
D[Log[Sqrt[x^2-1]],x]
      x
-----------
-1 + x^2
```

Il peut calculer la dérivée en remplaçant la variable par un nombre réel:

```
D[ArcTan[x^2+1],x]/.x->5
10
---
677
```

Il dérive des expressions symboliques qui peuvent servir de formules mathématiques:

```
D[ArcSin[a Cos[x]+b]^n,x]
                                    -1 + n
                    a n ArcSin[b + a Cos[x]]     Sin[x]
-(-------------------------------------------------------)
                                      2
                  Sqrt[1 - (b + a Cos[x]) ]
```

Il peut également calculer la dérivée seconde:

```
D[Sin[x],{x,2}]
-Sin[x]
```

```
D[ArcSin[x^3 + x^2 + 1], {x,2}]
```

$$\frac{(2x + 3x^2)^2 (1 + x^2 + x^3)}{(1 - (1 + x^2 + x^3)^2)^{3/2}} + \frac{2 + 6x}{\text{Sqrt}[1 - (1 + x^2 + x^3)^2]}$$

```
D[Log[1/Sqrt[x^2 + 1] + 1], {x,2}]
```

$$-\left(\frac{x^2}{(1 + x^2)^3 \left(1 + \frac{1}{\text{Sqrt}[1 + x^2]}\right)^2}\right) + \frac{3x^2}{(1 + x^2)^{5/2} \left(1 + \frac{1}{\text{Sqrt}[1 + x^2]}\right)} - \frac{1}{(1 + x^2)^{3/2} \left(1 + \frac{1}{\text{Sqrt}[1 + x^2]}\right)}$$

Il peut calculer la dérivée troisième, il faut indiquer cette dérivée par D[fonction,{x,3}]:

```
D[Sin[x], {x,3}]
```

```
-Cos[x]
```

```
D[ArcSin[x^3 + x^2 + 1], {x,3}]
```

$$\frac{3 (2x + 3x^2)^3 (1 + x^2 + x^3)^2}{(1 - (1 + x^2 + x^3)^2)^{5/2}} + \frac{(2x + 3x^2)^3}{(1 - (1 + x^2 + x^3)^2)^{3/2}} + \frac{3 (2 + 6x) (2x + 3x^2) (1 + x^2 + x^3)}{(1 - (1 + x^2 + x^3)^2)^{3/2}} + \frac{6}{\text{Sqrt}[1 - (1 + x^2 + x^3)^2]}$$

```
D[Log[1/Sqrt[x^2 + 1] + 1], {x, 3}]
```

$$\frac{-2\,x^3}{(1+x^2)^{9/2}\left(1+\frac{1}{\mathrm{Sqrt}[1+x^2]}\right)^3} + \frac{9\,x^3}{(1+x^2)^4\left(1+\frac{1}{\mathrm{Sqrt}[1+x^2]}\right)^2} - \frac{3\,x}{(1+x^2)^3\left(1+\frac{1}{\mathrm{Sqrt}[1+x^2]}\right)^2} - \frac{15\,x^3}{(1+x^2)^{7/2}\left(1+\frac{1}{\mathrm{Sqrt}[1+x^2]}\right)} + \frac{9\,x}{(1+x^2)^{5/2}\left(1+\frac{1}{\mathrm{Sqrt}[1+x^2]}\right)}$$

Exemple 3.3.3.

Montrer que la fonction $y = e^{-\alpha x}(a \cos x + b \sin x)$ vérifie l'équation $y'' + 2\alpha y' + (\alpha^2 + 1)y = 0$, pour toutes les valeurs de a et b.

```
Clear[f, a, b, alpha];
f[x_] := E^(-alpha x) (a Cos[x] + b Sin[x]);
f''[x] + 2 alpha f'[x] + (alpha^2 + 1) f[x]
```

$$\frac{-2\,\mathrm{alpha}\,(b\,\mathrm{Cos}[x] - a\,\mathrm{Sin}[x])}{E^{\mathrm{alpha}\,x}} + \frac{-(a\,\mathrm{Cos}[x]) - b\,\mathrm{Sin}[x]}{E^{\mathrm{alpha}\,x}} + \frac{\mathrm{alpha}^2\,(a\,\mathrm{Cos}[x] + b\,\mathrm{Sin}[x])}{E^{\mathrm{alpha}\,x}} + \frac{(1+\mathrm{alpha}^2)\,(a\,\mathrm{Cos}[x] + b\,\mathrm{Sin}[x])}{E^{\mathrm{alpha}\,x}} + 2\,\mathrm{alpha}\left(\frac{b\,\mathrm{Cos}[x] - a\,\mathrm{Sin}[x]}{E^{\mathrm{alpha}\,x}} - \frac{\mathrm{alpha}\,(a\,\mathrm{Cos}[x] + b\,\mathrm{Sin}[x])}{E^{\mathrm{alpha}\,x}}\right)$$

```
Simplify[%]
```

```
0
```

Dans ce dernier exemple il est important de vider toutes les variables avec la fonction **Clear[ ]**. S'il restait une valeur quelconque dans une des variables il serait impossible de vérifier que l'équation en y, y' et y" égale zéro. Le résultat pourrait être zéro pour d'autres raisons, par exemple, on sait que la dérivée d'une constante est nulle; le zéro pourrait être le résultat de la dérivée d'une valeur particulière.

3.4 L'applications de la dérivée.

L'interprétation géométrique et physique qu'on donne à la dérivée permet de retrouver cette notion dans plusieurs champs d'investigation: en géométrie, en physique, en chimie, en sociologie, en économique, etc. Les différents contextes donnent des interprétations différentes aux dérivées.

Exemple 3.4.1.

Dans le contexte géométrique, la dérivée en un point s'interprète comme la pente de la tangente en ce point.

Dans le premier quadrant, trouver l'équation de la tangente à l'ellipse $b^2x^2 + a^2y^2 = a^2b^2$, au point $x = c$.

Solution:

Pour trouver l'équation d'une droite il faut connaître la pente. Cette pente se calcule par la dérivée au point $x = c$.

```
Clear[f,a,b];

f[x_]:= b/a Sqrt[a^2 -x^2];

Print["y - ",f[c]," = ",f'[c]," (x - c)"]
```

L'équation de la droite est:

$$y - \frac{b\sqrt{a^2 - c^2}}{a} = -\left(\frac{b\,c}{a\sqrt{a^2 - c^2}}\right)(x - c)$$

Exemple 3.4.2

Dans le contexte géométrique, le signe de la dérivée seconde donne la concavité d'une fonction:

si la dérivée seconde est positive la concavité est vers le haut;

si la dérivée seconde est négative la concavité est vers le bas.

Soit la fonction $f(x) = e^{-1/x}$, donner le sens de la concavité au point x = 3.

```
Clear[f];
f[x_]:= E^(-1/x);
f''[3]
```

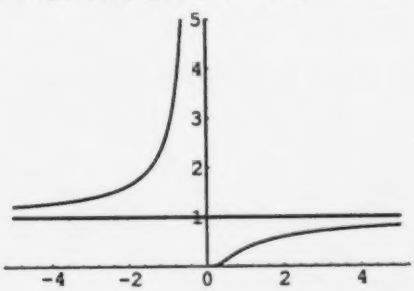
$$\frac{-5}{81\ E^{1/3}}$$

```
%//N
-0.0442303
```

Le signe de la dérivée seconde est négatif, on en conclut donc que la concavité à x = 3 est vers le bas.

Il est possible de faire la vérification à l'aide d'un graphe.

```
Plot[{1,f[x]},{x,-5,5},PlotRange->{0,5}]
```



Exemple 3.4.3.

Toujours dans le contexte géométrique, on peut utiliser les notions de pente et de concavité pour trouver le maximum ou le minimum relatif d'une fonction:

si la dérivée première est nulle et si la concavité est vers le haut la fonction passe par un minimum;

si la dérivée première est nulle et si la concavité est vers le bas la fonction passe par un maximum.

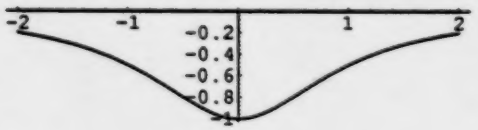
Remarque.- La fonction peut avoir un extrémum relatif pour d'autres raisons: soit que l'extrémum se trouve aux bornes du domaine de la fonction, soit que la dérivée n'existe pas, etc.

Trouver le minimum de la fonction $f(x) = -1/(x^2 + 1)$

```
Clear[f];

f[x_]:=-1/(x^2 + 1);

Solve[f'[x]==0,{x}]

{{x -> 0}}

f''[0]

2
```

Comme la dérivée première s'annule à x = 0, on vérifie le sens de la concavité en ce point. La dérivée seconde étant positive on conclut qu'à x = 0 la fonction passe par un minimum. Vérifions par un tracé de courbe.

```
Plot[{0,f[x]},{x,-2,2},AspectRatio->Automatic]
```



Exemple 3.4.4.

Dans le contexte géométrique on appelle point d'inflexion un point du graphe où il y a un changement de concavité.

Trouver les points d'inflexion du graphe $f(x) = x^3 / (x^3 - 1)$.

```
f[x_]:= x^3/(x^3 -1)

f'[x]
```

$$\frac{-3\,x^5}{(-1 + x^3)^2} + \frac{3\,x^2}{-1 + x^3}$$

```
Simplify[%2]
```

$$\frac{-3\,x^2}{1 - 2\,x^3 + x^6}$$

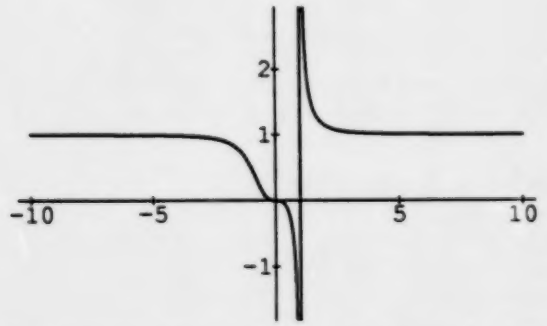
```
f''[x]
```

$$\frac{18\,x^7}{(-1+x^3)^3} - \frac{24\,x^4}{(-1+x^3)^2} + \frac{6\,x}{-1+x^3}$$

```
Simplify[%3]
```

$$\frac{6\,x\,(1+2\,x^3)}{(-1+x)^3\,(1+x+x^2)^3}$$

```
Plot[f[x],{x,-10,10}]
```

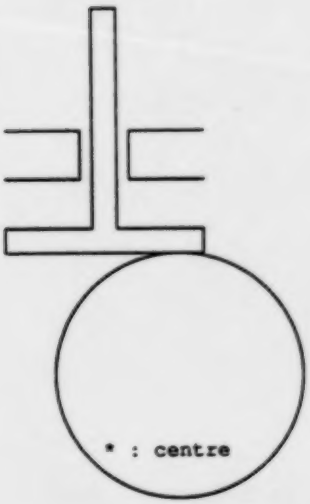


Cette fonction ne possède ni minimim, ni maximum relatif, cependant elle possède deux points d'inflexion à $x = 0$ et à $x = (-1/2)^{1/3}$.

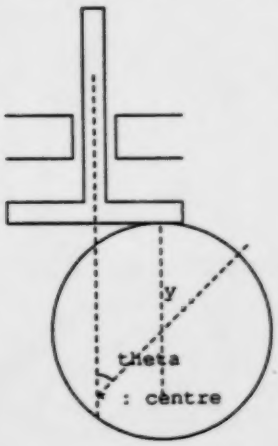
Exemple 3.4.5.

Dans un contexte physique, la dérivée en fonction de l'angle représente la vitesse angulaire.

Le came, illustré dans la figure qui suit, soulève et abaisse un culbuteur. La roue a un rayon de 7 cm . Le centre du came (représenté par un astérisque dans la figure) est à 5 cm du centre de la roue. Si la rotation du came est de 50 tours par minute quelle est la vitesse du culbuteur, quand l'angle de rotation est de $37°$?




```
exc=Graphics[Line[Table[{5 Cos[theta],5 Sin[theta]},
{theta,0,2Pi,.01}]]];
ad=Graphics[Line[{{1,8},{-2,8},{-2,10},{1,10}}]];

ag=Graphics[Line[{{-7,8},{-4,8},{-4,10},{-7,10}}]];
lb=Graphics[Line[{{-7,5},{1,5},{1,6},{-2.5,6},{-2.5,15},
{-3.5,15},{-3.5,6},{-7,6},{-7,5}}]];

Show[exc,lb,ad,ag,Graphics[Text["* : centre",{-3,-2.8},{-1,1}]]
,AspectRatio->Automatic]
```



```
exc=Graphics[Line[Table[{5 Cos[theta],5 Sin[theta]},
{theta,0,2Pi,.01}]]];
ad=Graphics[Line[{{1,8},{-2,8},{-2,10},{1,10}}]];
ag=Graphics[Line[{{-7,8},{-4,8},{-4,10},{-7,10}}]];
lb=Graphics[Line[{{-7,5},{1,5},{1,6},{-2.5,6},{-2.5,15},
{-3.5,15},{-3.5,6},{-7,6},{-7,5}}]];

lcp=Graphics[{Dashing[{0.02,0.02}],Line[{{-3,-3},{4,4}}]}];
lcc=Graphics[{Dashing[{0.02,0.02}],Line[{{0,-3},{0,5}}]}];
lccp=Graphics[{Dashing[{0.02,0.02}],Line[{{-3,-4},{-3,12}}]}];

Show[exc,lb,ad,ag,lcp,lcc,lccp,
Graphics[Text["* : centre",{-3,-2.8},{-1,1}]],
Graphics[Text["y",{0.2,2},{-1,1}]],
Graphics[Text["theta",{-2,-1},{-1,1}]],
Graphics[Circle[{-3,-3},1,{Pi/4,Pi/2}]]
,AspectRatio->Automatic]
```

Remarque.- Habituellement dans un texte ou un rapport on ne donne pas les fonctions qui tracent la figure. Il est toujours possible de garder la cellule qui contient la figure et de faire disparaître les cellules qui contiennent les fonctions. Comme le but du présent ouvrage est de montrer la souplesse du logiciel il est naturel de tout laisser. Il faut surtout prendre conscience que *Mathematica* dessine ou peut dessiner tous les schémas qui donnent plus de clarté à la compréhention d'un texte.

Solution:

À l'aide des constructions faites sur la figure ci-haut, on voit que y est égal à la somme du rayon du cercle plus la projection verticale de la distance entre les deux centres (cercle et came).
Donc, y (thêta) = 7 + 5 cos(thêta).

```
y[theta_]:=7 + 5 Cos[theta]
```

La vitesse s'interprète comme la dérivée au temps t dans un contexte physique:

```
Dt[y[theta],t]

-5 Dt[theta, t] Sin[theta]
```

50 tours par minute représente 100 π radians par minute.

La vitesse angulaire ou la dérivée de thêta par rapport au temps est de 100 π rad/min.
On cherche la vitesse de y quand thêta a tourné de 37°.

```
-5 100 Pi Sin[37 Degree]//N

-945.329
```

A 37° de la position d'origine le culbuteur descend à un vitesse de 945.329 cm/min.

Exemple 3.4.6.

Deux sources lumineuses sont à 12 mètres l'une de l'autre. Une d'entre elles est huit fois plus brillante que l'autre. L'intensité lumineuse est donnée par $L(x) = 8/x^2 + 1/(12 - x)^2$. Quel est le point situé entre les deux sources où l'intensité est la plus faible?

Remarque.- *L'intensité lumineuse est inversement proportionnelle au carré de la distance d'un des points à la source*. Ce qui explique l'équation L(x).

La solution est de trouver la dérivée, de l'annuler et de résoudre par rapport à x.

```
l[x_]:= 8/x^2 + 1/(12 - x)^2;

NSolve[l'[x]==0,x]

{{x -> 8.}, {x -> 12. - 6.9282 I}, {x -> 12. + 6.9282 I}}
```

Le seul nombre réel est x = 8. Les autres nombres sont des nombres complexes qui n'ont aucune signification physique dans ce contexte.

Une façon de vérifier ce résultat est de faire un tableau qui donne toutes les valeurs de la fonction entre 0 et 12 avec une gradation de .125. Et on utilise la fonction **Min[ ]** qui donne la plus petite valeur du tableau:

```
Min[Table[l[x],{x,.125,11.875,0.125}]]

0.1875
```

En prenant le minimum de ce tableau on trouve une valeur qui correspond à l'image de l(8):

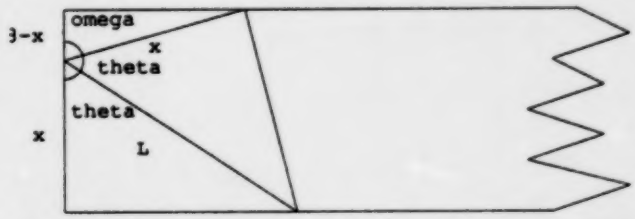
```
l[8.]

0.1875
```

Exemple 3.4.7.

On plie le coin d'une feuille de papier sur le bord opposé. La largeur de la feuille est 8 cm, la longueur du pli est L et la distance du coin originel au pli est x (voir figure ci-dessous). Exprimer L comme une fonction de x et trouver x qui minimise L.

Le tracé de la figure:

```
pli=Line[{{0,0},{0,8},{20,8},
{22,7},{19,6},{21,5},{18,4},{21,3},{18,2},{22,1},{19,0},
{0,0},{0,6},{7,8},{9,0},{0,6}}];
Show[Graphics[{pli,
              Text["  8-x",{-1.4,7}],Text["x",{-1,3}],
              Text["x",{3.5,6.5}],Text["L",{3,2.5}],
              Text["theta",{1.5,4}],Text["theta",{2.5,5.75}],
              Text["omega",{1.5,7.5}],
              Circle[{0,6},.75,{-Pi/2,Pi/2}]
              }],AspectRatio->Automatic]
```



La longueur de la feuille importe peu, elle doit avoir au moins 8 cm.
omega=Pi - 2 thêta;
Cos[thêta]=x / L;
Sin[thêta]=Sqrt[L^2 -x^2] / L;
Cos[oméga]=(8-x) / x;

```
Clear[omega,theta,x];
Solve[(8-x)/x==-((x/l)^2 -(Sqrt[l^2-x^2]/l)^2),{l}]
```

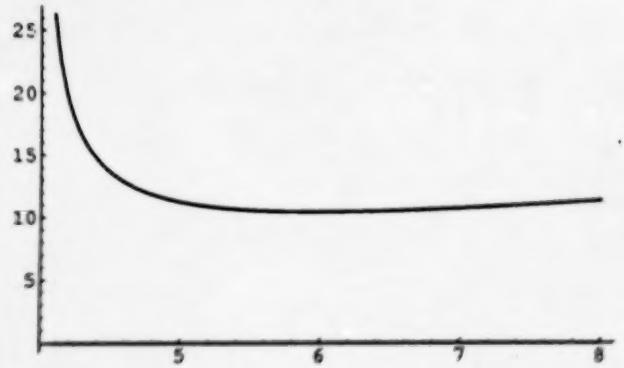
$$\{\{l \to \text{Sqrt}[16 - \frac{64}{4 - x} + 4\,x + x^2]\},\ \{l \to -\text{Sqrt}[16 - \frac{64}{4 - x} + 4\,x + x^2]\}\}$$

```
Simplify[%]
```

$$\{\{l \to \text{Sqrt}[\frac{x^3}{-4 + x}]\},\ \{l \to -\text{Sqrt}[\frac{x^3}{-4 + x}]\}\}$$

```
l[x_]:=Sqrt[x^3/(x-4)]

Solve[l'[x]==0,{x}]
{{x -> 6}}
```

```
Plot[{0,1[x]},{x,4.1,8}]
```



La fonction L atteint une valeur minimale au point x = 6. Ce résultat est trouvé par la dérivée et vérifié par le grahique.

Exemple 3.4.8.

D'après le principe de Fermat en optique, un rayon lumineux qui part d'un point P au-dessus de l'eau vers un point Q dans l'eau suit un trajet qui minimise le temps. Démontrer.

Soient:
thêta l'angle que le rayon fait dans l'air avec la verticale;
phi celui que le rayon fait dans l'eau avec la verticale ;

v_a la vitesse de la lumière dans l'air;

v_e la vitesse dans l'eau.

Montrer la loi de Snell: sin (thêta) / sin (phi) = v_a / v_e

Le reste de la solution est laissé en exercice.

3.5 Les fonctions paramétriques.

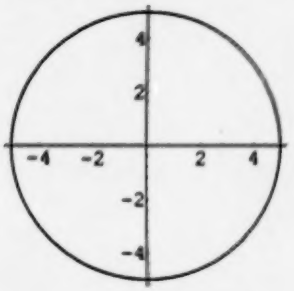
Il est souvent plus pratique de définir y indirectement comme une fonction de x au moyen de deux équations de la forme y = g(t) et x = h(t) où t est une troisième variable appelée paramètre. Les fonctions paramétriques remplacent d'une certaine façon les relations. Par exemple, la relation définie par le cercle $x^2 + y^2 = r^2$ donne généralement deux fonctions appelées fonctions demi-cercle. En équations paramétriques, il suffit d'écrire x = r cos t et y = r sin t.

Exemple 3.5.1.

À l'aide des équations paramétriques, tracer un cercle de rayon cinq.

Il existe dans *Mathematica* une fonction qui trace les fonctions paramétriques, **ParametricPlot[ ]**:

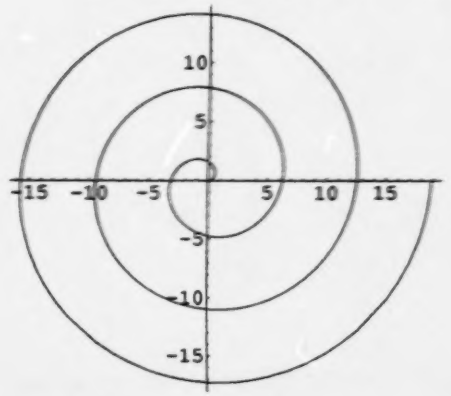
```
r = 5;
ParametricPlot[{r Cos[t],r Sin[t]},{t,0, 2 Pi},
AspectRatio -> Automatic]
```



Exemple 3.5.2.

La spirale d'Archimède est définie par: $x = t \cos(t)$ et $y = t \sin(t)$

```
ParametricPlot[{theta Cos[theta],theta Sin[theta]},
    {theta,0,6 Pi},AspectRatio->Automatic]
```




```
-Graphics-
```

Avec ces fonctions, les maximums, les minimums et les points d'inflexion se calculent comme les fonctions cartésiennes. On se sert de l'annulation de la dérivée première pour les maximums et les minimums et de celle de la dérivée seconde pour les points d'inflexion.

La dérivée première est définie par:

```
dy/dx  =  (dy/dt)*(dt/dx)
```

et la dérivée seconde par:

```
d^2y/dx^2  =  d/dx(dy/dx)  =  d/dt(dy/dx)/(dx/dt)
```

Exemple 3.5.3.

Pour trouver la dérivée première et la dérivée seconde de la spirale d'Archimède au point t = 3, il faut définir $x = t\ \cos(t)$ et $y = t\ \sin(t)$

```
Clear[a,x,y];

y[t_]:= t Cos[t];

x[t_]:= t Sin[t];
```

La dérivée première au point t = 3 est:

```
D[y[t],t]/D[x[t],t]/.t->3
```

$$\frac{\text{Cos[3] - 3 Sin[3]}}{\text{2 Cos[3] + Sin[3]}}$$

La fonction N[ ] nous donne la valeur numérique

```
N[%,5]
0.49962
```

La dérivée seconde au point t = 3 est:

```
(D[D[y[t],t]/D[x[t],t],t]/ D[x[t],t])/.t->3
```

$$\frac{-\left(\frac{(\text{Cos[3] - 3 Sin[3]})\ (\text{2 Cos[3] - 3 Sin[3]})}{(\text{3 Cos[3] + Sin[3]})^2}\right) + \frac{\text{-3 Cos[3] - 2 Sin[3]}}{\text{3 Cos[3] + Sin[3]}}}{\text{3 Cos[3] + Sin[3]}}$$

Sa valeur numérique est:

```
N[%,5]
0.48591
```

Exemple 3.5.4.

Nous allons trouver les dérivées première et seconde de la cycloïde dont les équations paramétriques sont:

```
Clear[x,y,t,a];

y[t_]:= a(1 - Cos[t]);

x[t_]:= a(t - Sin[t]);
```

La dérivée première est:

```
D[y[t],t]/D[x[t],t]
```

$$\frac{\text{Sin}[t]}{1 - \text{Cos}[t]}$$

La dérivée seconde est:

```
(D[D[y[t],t]/D[x[t],t],t])/ D[x[t],t]
```

$$\frac{\frac{\text{Cos}[t]}{1 - \text{Cos}[t]} - \frac{\text{Sin}[t]^2}{(1 - \text{Cos}[t])^2}}{a\ (1 - \text{Cos}[t])}$$

Nous pouvons simplifier cette dernière expression avec la fonction **Simplify[ ]**:

```
Simplify[%]
```

$$\frac{-\text{Csc}[\frac{t}{2}]^4}{4\ a}$$

Exemple 3.5.5.

Montrer qu'en tout point de la courbe $x = \ln(t)$, $y = t + 4$ la dérivée première égale la dérivée seconde.

Solution:

Nous devons d'abord enregistrer la nouvelle fonction dans *Mathematica*. Ensuite on calcule la dérivée première suivie de la dérivée seconde. On trace un graphe afin de visualiser les deux calculs.

```
Clear[t,x,y];
x[t_]:= Log[t];
y[t_]:= t + 4;
```

La dérivée première est:

```
D[y[t],t]/D[x[t],t]
```

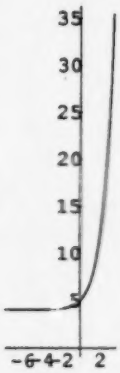
```
t
```

La dérivée seconde est:

```
(D[D[y[t],t]/D[x[t],t],t])/ D[x[t],t]
```

```
t
```

```
ParametricPlot[{x[theta],y[theta]},
    {theta,10^-10,10 Pi},AspectRatio->Automatic]
```

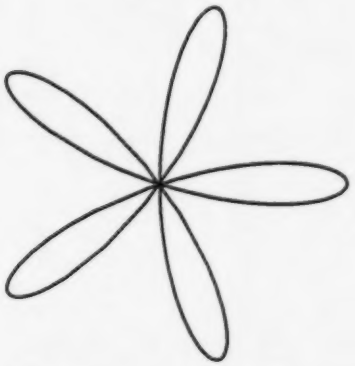


Exemple 3.5.6.

On peut facilement reproduire les graphiques des fonctions paramétriques usuelles.

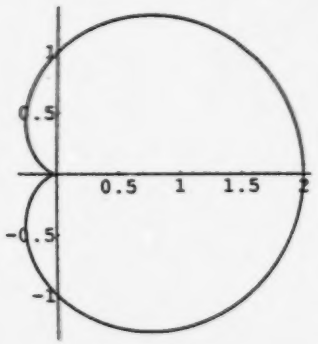
a) une rosace à cinq feuilles:

```
r[theta_]:= Cos[5 theta];

ParametricPlot[{r[theta] Cos[theta],r[theta] Sin[theta]},
          {theta,0,Pi},AspectRatio->Automatic,
          Axes->None]
```



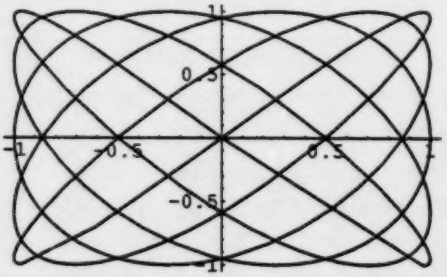
b) une cardioïde

```
r[theta_]:=1 + Cos[theta];

ParametricPlot[{r[theta] Cos[theta],r[theta] Sin[theta]},
          {theta,0,2 Pi},AspectRatio->Automatic]
```



c) une fonction digne d'un «Spirograph»

```
ParametricPlot[{Sin[5 t], Sin[6 t]},{t,0, 2Pi}]
```



Exercice 3.5.7.

Montrer que $d^2y/dx^2 = \det[\mathbf{m}] / (dx/dt)^3$, où **m** est la matrice $\{\{dx/dt, dy/dt\}, \{d^2x/dt^2, d^2y/dt^2\}\}$ et det[**m**] est le déterminant de **m**:

$$\begin{matrix} \frac{dx}{dt} & \frac{dy}{dt} \\ \frac{d^2x}{dt^2} & \frac{d^2y}{dt^2} \end{matrix}$$

▫ Chapitre 4.

4.1 Les différentielles.

L'équation $\lim_{x->0} (f(x+h) - f(x))/ h = f'(x)$ est équivalente à $f(x+h) - f(x) = h f'(x) + \varepsilon h$

ou bien $y + \Delta y = f(x + \Delta x) = f(x) + f'(x) \Delta x + \varepsilon \Delta x$, où ε est un petit nombre qui tend vers zéro quand $h = \Delta x$ tend aussi vers zéro. De cette équation nous pouvons écrire:

$\Delta y = f'(x) \Delta x + \varepsilon \Delta x$, i.e. que l'accroissement de la fonction est proportionnel à

$h = \Delta x$ additionné d'une petite quantité $\varepsilon \Delta x$ qui tend vers zéro quand Δx tend lui-même vers zéro. Ce sont ces idées qui ont conduit à la notion de différentielle. Tel que l'a souligné R. Courant dans son livre *Differential and Integral Calculus* l'idée de différentielle comme une quantité infiniment petite n'a aucun sens et il semble futile de définir la dérivée comme un quotient de telles quantités. Nous devons donc donner une définition précise de **dx** et de **dy**.

Définition: La quantité $h = \Delta x$ est la différentielle de x, notée **dx**;

l'expression **y'dx** est la différentielle de la fonction y = f(x), notée **dy**.

On a donc,

La différentielle de x est: $dx = \Delta x$;
la différentielle de y est: $dy = y'dx$;

Et par extension:

la seconde différentielle de y est: $d^2y = h^2f''(x)$;

la troisième différentielle de y est : $d^3y = h^3f'''(x)$;

. . . . . .

la $n^{ième}$ différentielle de y est: $d^ny = h^nf^{(n)}(x)$.

Il est à remarquer que pour le physicien, le biologiste, l'ingénieur, etc. la notion de dérivée vue comme un quotient de quantités physiques infinitésimales peut avoir un sens précis, si ces quantités sont choisies assez petites. Assez petites dans le sens de plus petite que le degré de précision d'une mesure qui donne la distance entre l'électron et le noyau d'un atome, par exemple.

Dans un travail scientifique on détermine la valeur d'une quantité **x** en effectuant des mesures sur cette quantité et on utilise cette valeur mesurée pour calculer la valeur d'une autre quantité **y**. Selon l'instrument utilisé il est possible d'estimer la plus grande erreur dans la mesure de **x**. Le scientifique s'intéresse à l'erreur correspondante dans le calcul de **y**. Cette partie des mathématiques s'appelle le calcul d'erreur. Le calcul des différentielles permet d'effectuer le calcul d'erreur. Voici quelques exemples:

Exemple 4.1.1.

Si nous trouvons que le diamètre d'une barre cylindrique est 5.7 cm avec une erreur de 0.05 cm, quelle est l'erreur possible dans le calcul de l'aire d'une des extrémités de la barre?

Solution:

La vraie mesure du diamètre se situe entre 5.65 et 5.75 cm. La plus grande erreur possible est alors la quantité avec laquelle l'aire du cercle croît quand son diamètre passe de 5.7 à 5.75 cm.

L'aire du cercle $A = \pi r^2 = \pi D^2/4$, où D est le diamètre du cercle.
dA nous donne une bonne approximation du changement d'aire:

```
Dt[Pi Dia^2 /4]/.{Dia->5.7,Dt[Dia]->0.05}
0.1425 Pi
%//N
0.447677
```

Nous trouvons donc 0.45 cm^2.

On peut définir l'**erreur relative** comme étant le quotient dA / A multiplié par 100:

```
(Dt[Pi Dia^2 /4]/Pi /Dia^2 4. 100)/.{Dia->5.7,Dt[Dia]->0.05}
1.75439
```

L'erreur relative est de 1.75 %.

Une autre façon d'obtenir l'**erreur relative** est de faire le calcul suivant:

```
100 Dt[Log[Pi Dia^2 /4]]/.{Dia->5.7,Dt[Dia]->0.05}
1.75439
```

Remarque.- La deuxième façon est basée sur le fait que la différentielle de **ln u** est **du / u**, qui intervient dans le calcul de l'**erreur relative**. La méthode est plus directe, donc plus rapide.

Exemple 4.1.2.

Une erreur de 0.5 % dans la mesure du rayon d'une sphère a comme conséquence une erreur de 1.5% dans le calcul de son volume. Vrai ou Faux?

Solution:

Le volume d'une sphère est 4/3 πr^3

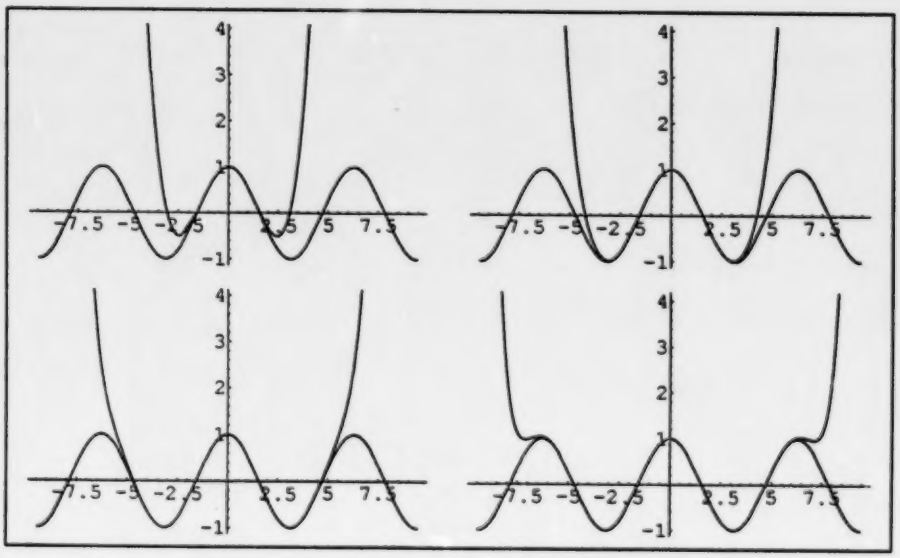
```
(Dt[4 Pi/3 r^3]/(4 Pi/3 r^3)100)/.Dt[r]->0.005 r
1.5
```

Ceci montre bien qu'une erreur de 0.5% dans la mesure du rayon induit une erreur de 1.5% dans le calcul du volume de la sphère. La réponse à la question est Vrai.

L'autre façon de faire le même calcul est de se servir de la fonction logarithmique:

```
100 Dt[Log[4 Pi/3 r^3]]/.Dt[r]->0.005 r
1.5
```

Les projets - Calcul II



Introduction.

Jusqu'à maintenant l'objectif principal était de trouver la dérivée d'une fonction. Dans plusieurs applications importantes du calcul le problème inverse se pose, il consiste à trouver la fonction qui donne une dérivée particulière. Trouver la longueur d'une courbe, l'aire bornée par des courbes, le travail fait par une force, le volume d'un objet géométrique, etc. demande la connaissance du procédé inverse de la dérivée, le calcul des primitives et des intégrales.

▫ Chapitre 1.

1.1 Les primitives et les intégrales

On appelle primitive d'une fonction f une fonction F telle que F' = f, i.e. si la dérivée de F donne la fonction f, alors F est une primitive de f. La primitive d'une fonction donnée n'est pas unique, en effet, il suffit d'ajouter une constante à F pour montrer que cette nouvelle fonction est aussi une primitive de f. *Mathematica* n'ajoute pas de constante aux primitives qu'on lui demande de calculer, cependant lorsqu'il travaille avec des équations différentielles les constantes sont ajoutées et il en tient compte dans les conditions initiales.

Exemple 1.1.1.

La fonction cos x a comme primitive la fonction sin x, car la dérivée de la fonction sin x est cos x.

```
Integrate[Cos[x],x]
```

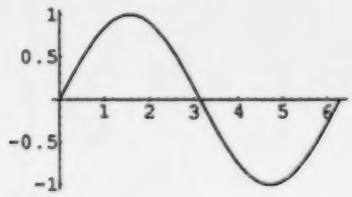
```
Sin[x]
```

Exemple 1.1.2.

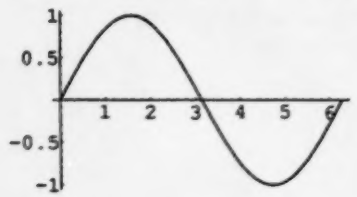
Considérons la fonction définie par:

```
f[x_]:=Integrate[Cos[t],{t,0,x}]
```

```
Plot[f[x],{x,0,2 Pi}]
```



```
Plot[Sin[u],{u,0,2 Pi}]
```

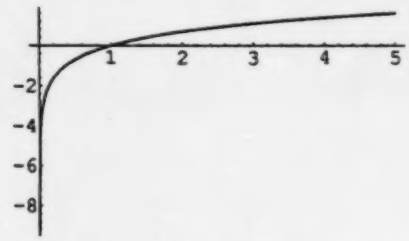


Les deux graphiques sont identiques. On peut donc se servir de la primitive d'une fonction pour définir de nouvelles fonctions. C'est le cas entre autre de la fonction logarithmique.

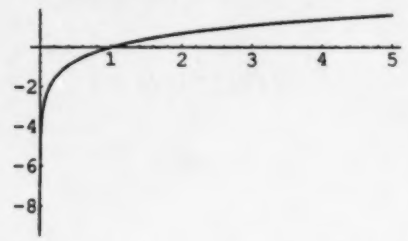
Exemple 1.1.3

L'exemple classique de la fonction logarithmique qui est définie comme étant la primitive de la fonction $f(t) = 1/t$ est:

```
lognat[x_]:=Integrate[1/t,{t,1,x}]

Plot[lognat[x],{x,0.0001,5}]
```




```
Plot[Log[x],{x,0.0001,5}]
```



Les deux graphiques sont identiques. Il est clair que la primitive $1/t$ est $\ln t$ et que la dérivée de $\ln t$ est $1/t$. C'est le même concept présenté différemment.

Exemple 1.1.4.

On démontre en mathématique que l'aire sous une courbe y = f(x) est une primitive de la fonction f. On peut donc se servir de l'intégrale définie pour faire des calculs d'aire.

L'aire algébrique bornée par la courbe $y = 18\,x / (x^2+5)$, l'axe des x et les verticales x = -1 et x = 7 est donnée par:

```
a = Integrate[18 x /(x^2 +5),{x,-1,7}]
-9 Log[6] + 9 Log[54]
N[a,5]
19.775
```

Si le système d'axes est mesuré en cm, par exemple, l'aire a est 19.775 cm^2.

Exemple 1.1.5.

Trouver l'aire comprise entre les deux courbes $y = 1/(x^2 + 4)$ et $y = x^2$.

Solution:

La première partie de la solution consiste à trouver les points de rencontre des courbes afin de déterminer les bornes d'intégration.

```
Solve[{1/(x^2+4)==x^2},{x}]
{{x -> (-2 + 5^(1/2))^(1/2)}, {x -> -(-2 + 5^(1/2))^(1/2)},
  {x -> (-2 - 5^(1/2))^(1/2)}, {x -> -(-2 - 5^(1/2))^(1/2)}}
```

On doit donc calculer l'aire entre les deux courbes entre `-(-2 + 5^(1/2))^(1/2)` et `(-2 + 5^(1/2))^(1/2)` ou deux fois l'aire entre 0 et `(-2 + 5^(1/2))^(1/2)`.

```
2 Integrate[1/(x^2 + 4) - x^2,{x,0,(-2 + 5^(1/2))^(1/2)}]
```

$$2\left(\frac{-(-2+\text{Sqrt}[5])^{3/2}}{3}+\frac{\text{Pi}}{4}-\frac{\text{ArcTan}\left[\frac{2}{\text{Sqrt}[-2+\text{Sqrt}[5]]}\right]}{2}\right)$$

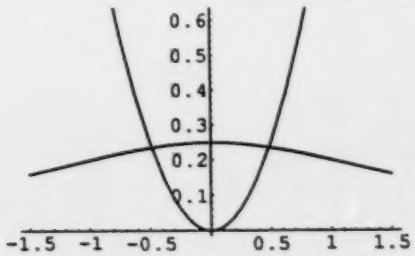
```
N[%,5]
0.16185
```

On peut faire la vérification avec une intégrale numérique:

```
2 NIntegrate[1/(x^2 + 4) - x^2,{x,0,(-2 + 5^(1/2))^(1/2)}]
0.161852
```

Le graphique qui suit nous montre que la symétrie peut être utilisée dans le calcul de l'aire entre les deux courbes.

```
Plot[{1/(x^2 + 4) , x^2},{x,-1.5,1.5}]
```



Exemple 1.1.6.

Trouver l'aire entre la courbe $y = -2/27\ x^3$ et la tangente à la courbe au point $x = -4$.

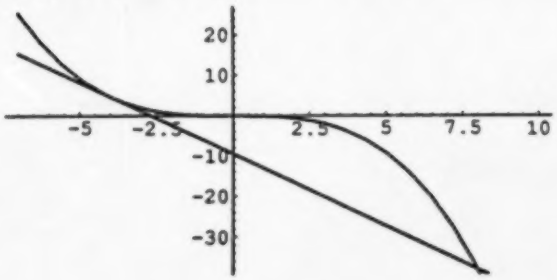
Solution:

Dans ce problème les deux concepts, dérivée et intégrale, se retrouvent. On doit se servir de la dérivée pour trouver l'équation de la tangente et de l'intégrale pour calculer l'aire entre les courbes.

```
f[x_]:= -2/27 x^3;

a=-4;

Solve[f[x] == f'[a](x-a)+f[a],{x}]
{{x -> 8}, {x -> -4}, {x -> -4}}
```

La présence d'une racine double dans les résultats donnés par la fonction **Solve[ ]** caractérise le fait que la courbe est tangente à la droite, comme nous le montre le graphique suivant:

```
Plot[{f[x],f'[a](x-a)+f[a]},{x,-7,10},
AspectRatio->1/2]
```



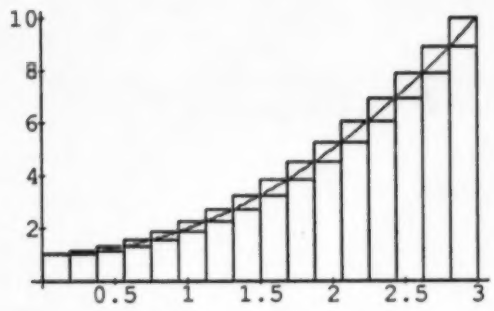

```
Integrate[-f[x] + f'[a](x-a)+f[a],{x,-2a,a}]
```

```
128
```

Exemple 1.1.7.

Voici un programme qui trace des rectangles par valeurs supérieures et inférieures. La somme des aires des rectangles donne une approximation de plus en plus précise de l'aire sous la courbe. Il sert à illustrer l'intégrale selon Riemann. On peut également s'en servir pour calculer approximativement une intégrale avec la somme des aires de quelques rectangles..

```
Clear[da,n,f,ff,k,r];
Do[da=3/n;
f[x_]:= 1 + x^2;
ff=Plot[f[x],{x,0,3},PlotStyle->RGBColor[1,0,1],
          AspectRatio->1/GoldenRatio];
rech[k_]:=Graphics[Line[{{k da,0},{k da,f[k da]},
      {(k+1)da,f[k da]},{(k+1)da,0},{k da,0}}]];
recb[k_]:=Graphics[Line[{{k da,0},{k da,f[(k+1) da]},
      {(k+1)da,f[(k+1) da]},{(k+1)da,0},{k da,0}}]];
Show[Table[{rech[r],recb[r]},{r,0,n-1}],ff];,{n,3,18}]
```



Remarque.- On peut changer la fonction et le nombre de rectangles dans le programme fait plus haut. Mais si nous reproduisons tel quel le programme nous obtenons une suite de graphiques qui peuvent être animés.

1.2 Les intégrales numériques.

Il peut arriver qu'une fonction donnée n'ait pas de primitives. Nous pouvons calculer son intégrale définie par des méthodes dites numériques. L'exemple suivant nous fournit un aperçu des diffférents modes de calculs d'une intégrale définie.

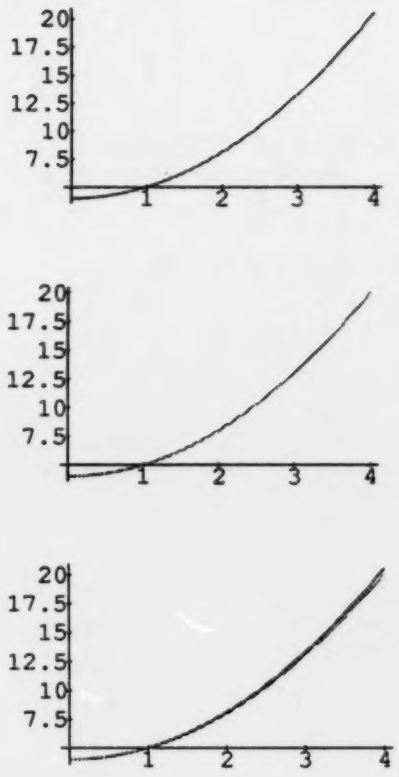
Exemple 1.2.1.

L'intégrale au sens de Reimann.

a) Si on définit la fonction somme de RIEMANN et si on la dérive, on retrouve la fonction qui a servi à composer la fonction de RIEMANN. On en déduit donc que la fonction somme est une primitive de la fonction de base.

fr est la fonction somme.
fi représente la dérivée de la fonction somme.
fs est la fonction elle-même pour fin de comparaison.
On voit que la fonction somme est une primitive de f[x]=x^2+4.

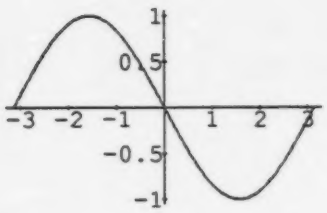
```
Clear[x,n,i,fi,fs,fr]
f[x_]:=x^2+4
n=50
fr[x_]:=Sum[f[i x/n] x/n,{i,1,n}]
fi=Plot[fr'[x],{x,0,4},PlotStyle->RGBColor[1,.1,1]]
fs=Plot[f[x],{x,0,4},PlotStyle->RGBColor[.2,1,.3]]
Show[fi,fs]
```

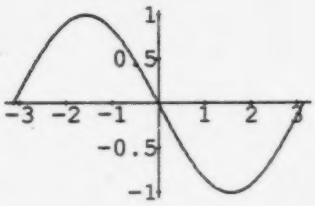


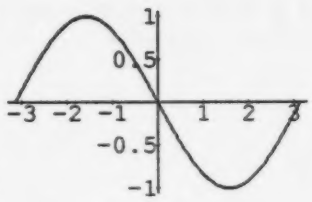
-Graphics-

b) La dérivée de la fonction intégrale donne la fonction intégrante.

```
f[x_]:=Integrate[-Sin[t],{t,Pi/2,x}]
fc=Plot[f'[x],{x,-Pi,Pi},PlotStyle->RGBColor[1,0,.5]]
fC=Plot[-Sin[x]-0.01,{x,-Pi,Pi},PlotStyle->RGBColor[0,.5,.5]]
Show[fc,fC]
```





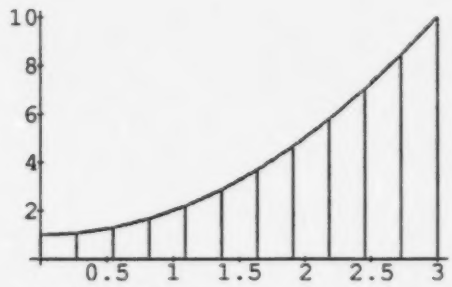


-Graphics-

c) La méthode des Trapèzes pour le calcul d'une intégrale.

```
Clear[da,n,f,ff,k,r]
Do[da=3/n;
f[x_]:= 1 + x^2;
ff=Plot[f[x],{x,0,3},PlotStyle->RGBColor[1,0,1],
                                    AspectRatio->1/GoldenRatio];
```

```
rec[k_]:=Graphics[Line[{{k da,0},{k da,f[k da]},
                {(k+1)da,f[(k+1) da]},
                {(k+1)da,0},{k da,0}}]];
Show[Table[rec[r],{r,0,n-1}],ff];,{n,3,18}]
```



d) La méthode de SIMPSON

i) Si f(x) est une fonction polynomiale de degré inférieur à 4, alors la valeur de l'intégrale par la méthode de Simpson donne la valeur exacte de l'intégrale, quelle que soit la valeur de n.
On utilise la valeur n = 2 dans ce cas.

ii) Pour un n pair quelconque avec une fonction quelconque:

```
Clear[f,a,b,n,prt,s,m]
  simp[a_,b_,n_]:=
   (prt=Table[a+i,{i,0,b-a,(b-a)/n}];
   s=N[(b-a)/(3n)   (f[a]+
      4   Sum[f[prt[[2m]]],{m,1,n/2}]+
      2   Sum[f[prt[[2m-1]]],{m,2,n/2}]+
        f[b]);
    ])
```

```
f[x_]:=Cos[x]/(1 + Sin[x]^2)
    simp[0,Pi/4,500]
```

```
N[%,20]
```

```
0.61547970886484904307
```

```
N[Integrate[Cos[x]/(1 + Sin[x]^2),{x,0,Pi/4}],20]
```

```
0.61547970867038734107
```

```
f[x_]:= 4/(1+x^2)
     simp[0,1,200]

N[%,15]

3.14159265358979

f[x_]:= 1/(2+Sin[x]^2)
     simp[0,Pi,4]

%

 73 Pi
 ---
 180

N[%,15]

1.27409035395586

f[x_]:= Sqrt[1+Sin[x]^2]
     simp[0,Pi,20]

N[%,15]

3.82019778953933

NIntegrate[Sqrt[1+Sin[x]^2],{x,0,Pi}]

3.820197787830711456

f[x_]:=(1+x)^x
     simp[0,3,50]

N[%,7]

33.01193

NIntegrate[(1+x)^x,{x,0,3}]

33.01187678022750454
```

Ce qui précède nous a montré une suite d'exemples avec vérification.

▫ Chapitre 2.

2.1 Les séries.

Mathematica identifie ses suites finies par des tables.

Exemple 2.1.1.

La suite harmonique est:

```
Table[1/n,{n,1,15}]
```

$$\{1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \frac{1}{6}, \frac{1}{7}, \frac{1}{8}, \frac{1}{9}, \frac{1}{10}, \frac{1}{11}, \frac{1}{12}, \frac{1}{13}, \frac{1}{14}, \frac{1}{15}\}$$

Et ses séries finies par des sommes:

```
Sum[1/n,{n,1,15}]
```

$$\frac{1195757}{360360}$$

Nous pouvons nous servir de ce fait pour définir le terme général d'une série infinie:

```
S[n_]:=Sum[1/i,{i,1,n}]
```

Voici la suite $(S_i)_n$ développée pour une vingtaine de termes:

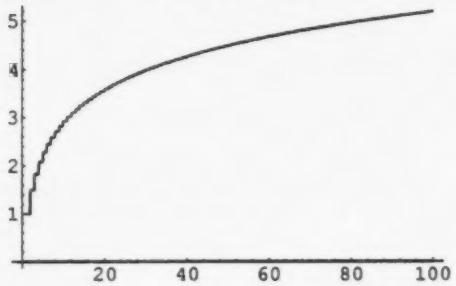
```
Table[S[w],{w,1,20}]
```

$$\{1, \frac{3}{2}, \frac{11}{6}, \frac{25}{12}, \frac{137}{60}, \frac{49}{20}, \frac{363}{140}, \frac{761}{280}, \frac{7129}{2520}, \frac{7381}{2520}, \frac{83711}{27720}, \frac{86021}{27720},$$
$$\frac{1145993}{360360}, \frac{1171733}{360360}, \frac{1195757}{360360}, \frac{2436559}{720720}, \frac{42142223}{12252240}, \frac{14274301}{4084080},$$
$$\frac{275295799}{77597520}, \frac{55835135}{15519504}\}$$

```
%//N
```

```
{1., 1.5, 1.833333333333333333, 2.083333333333333333,
  2.283333333333333333, 2.45, 2.592857142857142857,
  2.717857142857142857, 2.828968253968253968,
  2.928968253968253968, 3.019877344877344877,
  3.103210678210678211, 3.180133755133755134,
  3.251562326562326562, 3.318228993228993229,
  3.380728993228993229, 3.439552522640757935,
  3.495108078196313491, 3.547739657143681911,
  3.597739657143681911}
```

```
Plot[{0,S[n]},{n,1,100}]
```




```
-Graphics-
```

Exemple 2.1.2.

On donne la suite : {0, 3, 16, 45, 96, 175, 288, 441, ...} et on demande le terme général et le neuvième terme.

Solution:

Après avoir fait les différences des termes de la suite sur trois niveaux de profondeur on se rend compte que le terme général a la forme d'un polynôme du troisième degré.

```
p[n_]:= a n^3 + b n^2 + c n + d

Solve[{p[1]==0,p[2]==3,p[3]==16,p[4]==45},{a,b,c,d}]

{{a -> 1, b -> -1, c -> -1, d -> 1}}

Factor[p[n]/.%]

                    2
{(-1 + n)  (1 + n)}

%/.n->9

{640}
```

Bref,

le terme général est: $(n - 1)^2 (n + 1)$;

le neuvième terme est: 640.

2.2 Les séries infinies convergentes

Il est clair que si nous additionnons des termes positifs on s'attend d'obtenir une série croissante. Certaines de ces séries divergent, i.e. qu'elles deviennent plus grandes que n'importe quel nombre positif, aussi grand soit-il.

Exemple 2.2.1.

Soit la série 1 + 1/Sqrt[2] + 1/Sqrt[3] + 1/Sqrt[4] + ... + 1/Sqrt[n] + ...
Nous voyons immédiatement que S_n > 1/Sqrt[n] + 1/Sqrt[n] + 1/Sqrt[n] + ...+ 1/Sqrt[n] = n/Sqrt[n] = Sqrt[n].

$\text{Lim}_{n\to\infty}$ Sn > $\lim_{n\to\infty}$ Sqrt[n] = ∞.

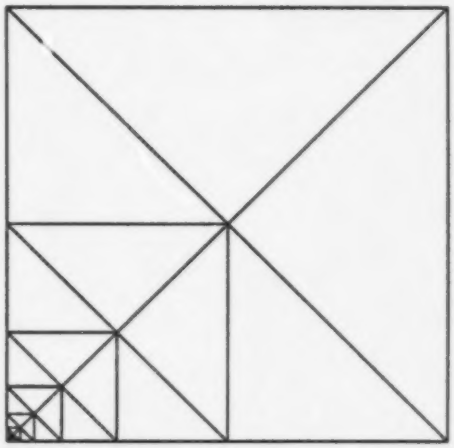
La $n^{ième}$ somme partielle croît au delà de toutes limites quand n tend vers l'infini. Donc la série diverge.

Exemple 2.2.2.

Parmi ces séries à termes positifs il y en a qui convergent, i.e. qu'elles s'approchent d'un nombre réel quand n tend vers ∞.

Le graphique qui suit va nous aider à visualiser de telles séries.

```
car[n_]:= Graphics[Line[{{n,0},{0,0},{0,n},{n,n},{n,0},
                         {0,0},{n,n},{0,n},{n,0}}]];
caremb = Table[car[1/2^i],{i,6,0,-1}];
Show[caremb,PlotRange->All,AspectRatio->Automatic]
```



Analysons attentivement le graphique ci-dessus. Au départ on a un carré de coté un, de la rencontre des diagonales on abaisse deux perpendiculaires qui forment un deuxième carré de coté 1/2. En répétant le même procédé on trace un troisième carré de coté 1/4 et ainsi de suite. On obtient donc une suite emboîtée de carrés de coté 1/2, 1/4,1/8,1/16, 1/32, ... Tous ces carrés sont contenus dans le carré de coté un. Comme le coté de l'un des carrés est la moitié de l'autre, il est donc naturel que 1/2 + 1/4 + 1/8 + 1/16 + 1/32 + ... = 1.
D'où, certaines séries à termes positifs sont convergentes, même si elles contiennent un nombre infini de termes.

Exemple 2.2.3.

La somme d'une série géométrique est $a\,(1 - q^n) / (1 - q)$, où a est le premier terme de la suite géométrique et q la raison.

Les séries géométriques sont des séries convergentes, si la raison q est $|q| < 1$, c'est-à-dire $-1 < q < 1$.

Si la série est convergente, la somme est $a / (1 - q)$.

Montrer que la série suivante est convergente et trouver la somme.

$1/2 + 1/6 + 1/18 + \ldots + 1/(2 * 3^n) + \ldots$

La somme est $a / (1 - q)$:

$$\frac{1}{2\ (1\ -\ \frac{1}{3})}$$

$$\frac{3}{4}$$

□ Chapitre 3.

3.1 Les séries de MacLAURIN.

Exemple 3.1.1.

Le polynôme $f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_6 x^6 + a_7 x^7$ est dérivé six fois de suite. À chaque fois on calcule l'image de chacune des dérivées à $x = 0$. On obtient $f(0) = a_0$, $f'(0) = a_1$, $f''(0) = 2\,a_2$, $f'''(0) = 6\,a_3$, $f^{(4)}(0) = 24\,a_4$, $f^{(5)}(0) = 120\,a_5$, $f^{(6)}(0) = 720\,a_6$, $f^{(7)}(0) = 5040\,a_7$, ...

Ce qui peut s'écrire:

$f(0) = a_0$, $f'(0) = a_1$, $f''(0) = 2!\,a_2$, $f'''(0) = 3!\,a_3$, $f^{(4)}(0) = 4!\,a_4$, $f^{(5)}(0) = 5!\,a_5$, $f^{(6)}(0) = 6!\,a_6$, $f^{(7)}(0) = 7!\,a_7$, ...

On peut isoler les a_i pour obtenir:

$a_0 = f(0)$, $a_1 = f'(0)$, $a_2 = f''(0)/2!$, $a_3 = f'''(0)/3!$, $a_4 = f^{(4)}(0)/4!$, $a_5 = f^{(5)}(0)/5!$, $a_6 = f^{(6)}(0)/6!$, $a_7 = f^{(7)}(0)/7!$ ...

Mathematica vérifie ce qui vient d'être fait:

```
c = Table[Subscripted[a[j]],{j,0,8}];

f[x_]:= Sum[c[[i]] x^(i-1),{i,1,8}];

Do[Print[D[f[x],{x,k}]/.x->0],{k,0,7}]
```

```
a
 0
a
 1
2 a
   2
6 a
   3
24 a
   4
120 a
   5
720 a
   6
5040 a
   7
```

Donc, pour une fonction multi-dérivable il est possible de trouver un polynôme qui coïncide avec cette fonction, du moins dans le voisinage de zéro.

Exemple 3.1.2.

Voyons un exemple avec la fonction sinus.

```
Clear[f]
Table[D[Sin[x],{x,k}]/.x->0,{k,0,12}]
{0, 1, 0, -1, 0, 1, 0, -1, 0, 1, 0, -1, 0}
Table[1/k! D[Sin[x],{x,k}]/.x->0,{k,0,12}]
```

$$\{0,\ 1,\ 0,\ -\left(\frac{1}{6}\right),\ 0,\ \frac{1}{120},\ 0,\ -\left(\frac{1}{5040}\right),\ 0,\ \frac{1}{362880},\ 0,\ -\left(\frac{1}{39916800}\right),\ 0\}$$

Voici les coefficients; les a_i du polynône: $0 + 1\,x + 0\,x^2 - 1/6\,x^3 + 0\,x^4 + 1/120\,x^5 + 0\,x^6 \cdots$
Mathematica donne la fonction sinus développée en série de MacLAURIN:

```
Normal[Series[Sin[x],{x,0,12}]]
```

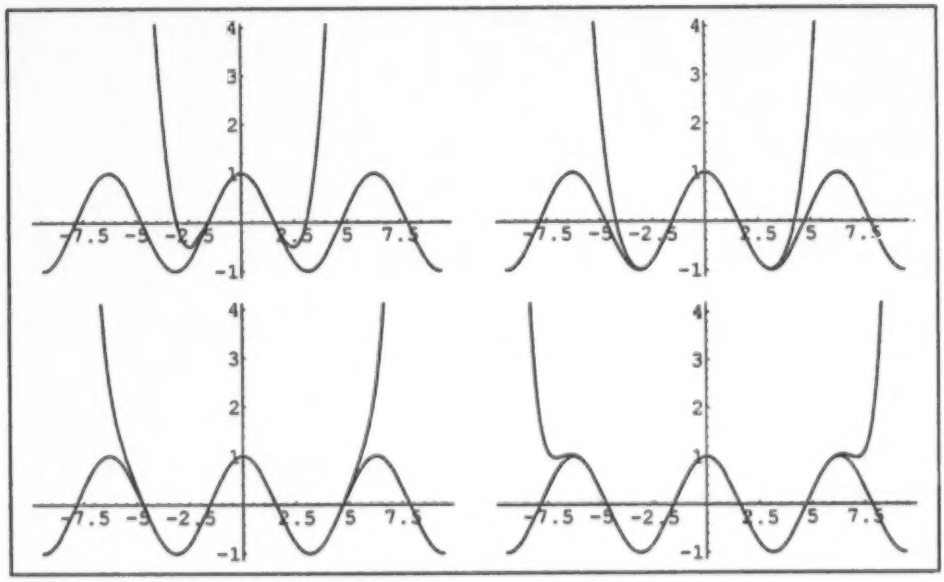
$$x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{5040} + \frac{x^9}{362880} - \frac{x^{11}}{39916800}$$

Exemple 3.1.3.

Concernant la fonction cosinus:

```
Clear[f,g,n,x,y];

f[x_,y_]:=Sum[(-1)^n x^(2 n)/(2 n)!,{n,0,y}];

g=Table[Plot[{Cos[x],f[x,y]},{x,-3Pi,3Pi},
    {y,2,8,2}];

Show[GraphicsArray[{{g[[1]],g[[2]]},{g[[3]],g[[4]]}}]]
```

```
Show[%,Frame->True]
```



On voit que plus on ajoute de termes à la série de MacLAURIN plus la fonction polynômiale coïncide avec la fonction cosinus.

Mathematica transforme en séries des fonctions composées très complexes où il serait très difficile, voire impossible de faire le calcul avec un cayon et un papier.

Exemple 3.1.4.

Dans ce qui suit nous verrons des exemples de fonctions et de remplacements possibles dans *Mathematica*:

a)

```
Normal[Series[Tan[x],{x,0,12}]]
```

$$x + \frac{x^3}{3} + \frac{2\,x^5}{15} + \frac{17\,x^7}{315} + \frac{62\,x^9}{2835} + \frac{1382\,x^{11}}{155925}$$

b)

```
Normal[Series[ArcTan[t],{t,0,12}]]/.t->Sqrt[x]
```

$$\text{Sqrt}[x] - \frac{x^{3/2}}{3} + \frac{x^{5/2}}{5} - \frac{x^{7/2}}{7} + \frac{x^{9/2}}{9} - \frac{x^{11/2}}{11}$$

c)

```
Normal[Series[Exp[Cos[Sin[x]]],{x,0,12}]]
```

$$E - \frac{E\,x^2}{2} + \frac{E\,x^4}{3} - \frac{127\,E\,x^6}{720} + \frac{3523\,E\,x^8}{40320} - \frac{12247\,E\,x^{10}}{302400} + \frac{2846159\,E\,x^{12}}{159667200}$$

b)

```
Normal[Series[ArcTan[t],{t,0,12}]]/.t->Sqrt[x]
```

$$\text{Sqrt}[x] - \frac{x^{3/2}}{3} + \frac{x^{5/2}}{5} - \frac{x^{7/2}}{7} + \frac{x^{9/2}}{9} - \frac{x^{11/2}}{11}$$

c)

```
Normal[Series[Exp[Cos[Sin[x]]],{x,0,12}]]
```

$$E - \frac{E\,x^2}{2} + \frac{E\,x^4}{3} - \frac{127\,E\,x^6}{720} + \frac{3523\,E\,x^8}{40320} - \frac{12247\,E\,x^{10}}{302400} + \frac{2846159\,E\,x^{12}}{159667200}$$

Chapitre 4.

4.1. Les opérations avec les séries de puissances.

Les opérations que l'on peut faire avec un nombre fini de termes ne sont pas toujours permises quand des séries infinies sont impliquées.

Addition.- Si deux séries, qui convergent respectivement vers U et V, sont additionnées (multipliées) la série résultante converge vers U + V (U V). La même chose se produit avec les séries de puissances ou de fonctions.

Différenciation et intégration.- À l'intérieur de l'intervalle de convergence on peut dériver ou intégrer terme à terme une fonction développée en série de puissances.

Exemple 4.1.1.

Évaluer l'aire sous la courbe de $f(x) = e^{-x^2}$ entre 0 et 1/2.

Il est impossible de trouver une primitive de cette fonction, mais par le développement en séries on peut calculer cette aire avec une très grande précision.

Solution:

```
Normal[Series[Exp[x],{x,0,30}]]/.x->-x^2;
Integrate[%,{x,0,1/2}]
```

$$\frac{42403204675238834230332247085156504036723131442827328871239786089229}{91924887618921241184868880736227753532019576460405823011553280000000}$$

```
N[%,20]
0.46128100641279244876
```

On peut effectuer une vérification avec la fonction **NIntegrate[ ]**:

```
NIntegrate[Exp[-x^2],{x,0,1/2}]
0.461281
```

Exemple 4.1.2.

Calculer l'intégrale de la fonction $y = \sin(x^3)\, e^{x^2}$ entre 0 et 1.

Solution:

```
Normal[Series[Sin[x^3] Exp[x^2],{x,0,35}]];
Integrate[%,{x,0,1}]
```

$$\frac{1184539354093703}{2560949482291200}$$

```
N[%,10]
```

```
0.4625391334
```

Vérification:

```
NIntegrate[Sin[x^3] Exp[x^2],{x,0,1}]
```

```
0.462539
```

Exemple 4.1.3.

Calculer l'intégrale de la fonction y = Sqrt[x] Cos[x] entre 0 et 1.

```
Normal[Series[Sqrt[x] Cos[x^2],{x,0,35}]];
Integrate[%,{x,0,1}]
```

$$\frac{111051805884961244377}{191454745577711616000}$$

```
N[%,10]
```

```
0.5800420645
```

Vérification:

```
NIntegrate[Sqrt[x] Cos[x^2],{x,0,1}]
```

```
0.580042
```

Exemple 4.1.4.

On peut se servir des séries pour calculer certaines limites:

Calculer la limite de la fonction $f(x) = \{ e^{x^3} + e^{-x^2} - 2\cos(4x) \} / x \sin(5x)$.

Solution:

On développe chacune des fonctions en séries:

```
s1 = Series[Exp[x^3],{x,0,10}];
s2 = Series[Exp[-x^2],{x,0,10}];
s3 = Series[Sin[5 x], {x,0,10}];
s4 = Series[Cos[4 x], {x,0,10}];
```

Après on calcule la limite:

```
Limit[(s1 + s2 - 2 s4)/( x s3),x->0]

3
```

Vérification:

```
Limit[(Exp[x^3] + Exp[-x^2] - 2 Cos[4 x])/(x Sin[5 x]),x->0]

3
```

Chapitre 5.

Le calcul de π se fait avec les séries quoique d'autres méthodes peuvent être utilisées si la précision exigée n'est pas très grande. Quoiqu'il en soit nous allons faire un essai avec la fonction arctan(x). Nous verrons que cette série converge très lentement et que nous devrons trouver un développement plus adéquat.

L'idée de base vient du fait que arctan(1) donne $\pi/4$. En développant 4 arctan(1) on doit trouver la valeur de π.

Première tentative:

```
Normal[Series[ArcTan[x],{x,0,5000}]]/.x->1.;
pi= 4 %
```

```
3.14119
```

Après 5000 termes on obtient une piètre approximation du nombre π! Nous allons prendre une autre stratégie de façon à diminuer le temps de convergence.

De la trigonométrie nous savons que:
tan(u - v) = (tan u - tan v) / (1 + tan u tan v).

Posons:
tan u = p et u = arctan p;
tan v = r et v = arctan r et prenons l'arctan( ) des deux membres:

arctan p - arctan r = arctan((p - r) / (1 + p r)).

La lenteur de la convergence vient du fait que le calcul se fait en remplacant x par **un** dans la série de MacLAURIN et on sait que **un** à n'importe quelle puissance donne toujours **un**. De la formule arctan p - arctan r = arctan((p - r) / (1 + p r)) nous pouvons partager ce **un** en deux fractions. Ces deux fractions assureront une convergence beaucoup plus rapide, car une fraction élevée à une puissance positive donne une fraction plus petite que la fraction originale. Pour le partage du **un** nous devons résoudre un système d'équations où p prend une valeur arbitraire, disons 1/12 et où (p - r) / (1 + p r) devra être égal à **un**.

```
Clear[p,r];
Solve[{p - r == 1 (1 + p r),p == 1/12},{r}]
```

$$\{\{r \to -(\frac{11}{13})\}\}$$

Deuxième tentative:

En s'inspirant du développement en série de l'arctan(x), posons la fonction
f(x) = Sum[(-1)^(n+1) x^(2n-1)/(2n-1),{n,1,50}]:

```
Clear[f];

f[x_]:= Sum[(-1)^(n+1) x^(2n-1)/(2n-1),{n,1,50}];

ppi = 4 N[(f[1/12] + f[11/13]),50]

3.1415926524954057114611300399627110639130641 9929
```

Vérification avec la valeur de Pi de *Mathematica* :

```
N[Pi-ppi,50]

1.0943875270015303792832317778058629873822 10^-9
```

La valeur de ppi est exacte pour au moins neuf chiffres significatifs. Nous allons affiner notre méthode en partageant, en deux parties, la fraction 11/13 , qui est trop près de un.

Troisième tentative:

```
Clear[p,r];
Solve[{p - r == 11/13 (1 + p r),p == 1/3},{r}]

{{r -> -(2/5)}}

Clear[f];

f[x_]:= Sum[(-1)^(n+1) x^(2n-1)/(2n-1),{n,1,50}];

pppi = 4 N[(f[1/12] + f[1/3]+ f[2/5]),50]

3.1415926535897932384626433832795028841971671989454
```

Vérification avec la valeur de Pi:

```
N[Pi-pppi,50]

2.2004297 10^-42
```

Une très nette amélioration sur l'exemple précédent, pppi donne 42 chiffres significatifs valables.

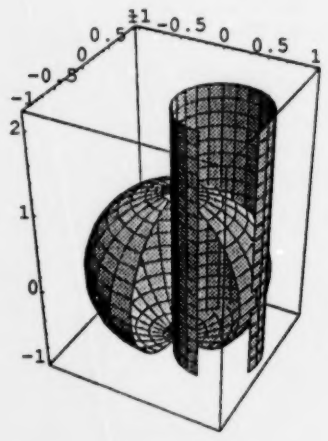
Afin d'améliorer davantage la précision de π, nous avons été amenés à découvrir la fonction qui suit, en utilisant un procédé analogue:

```
Clear[f];

f[x_]:= Sum[(-1)^(n+1) x^(2n-1)/(2n-1),{n,1,100}];

pi = N[(16 f[1/12] +  8 f[1/3]- 4 f[1127/5839]),100]

3.14159265358979323846264338327950288419716939937510582097\
  4459230781640628620899862803482534211702 3
```

Cette dernière fonction donne le nombre π avec 98 chiffres significatifs exacts.

Remarque.- Aujourd'hui l'utilité des séries n'est plus seulement de trouver la valeur précise de constantes célèbres, mais plutôt de résoudre des équations différentielles à coefficients variables. De ces calculs on a vu naître une panoplie de concepts dont les principaux permettent de résoudre les équations de Legendre, Bessel et Gauss.

Les projets - Calcul III



Introduction.

Mathematica possède toutes les fonctions nécessaires pour dériver et intégrer les fonctions à plusieurs variables. En plus d'un nombre impressionnant de ces fonctions *Mathematica* permet d'en créer d'autres à notre convenance ou selon notre besoin. Sa principale caractéristique, il va sans dire, est son option graphique en 3D, qui permet de visualiser et d'interprèter ces fonctions. Ce qui en facilite grandement le calcul.

▫ Chapitre 1.

1.1 Les fonctions hyperboliques.

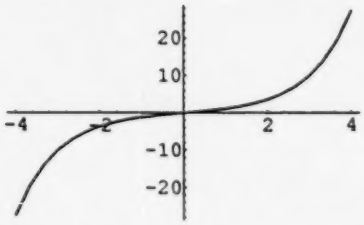
Les fonctions hyperboliques sont utilisées pour simplifier les résultats obtenus dans la solution d'équations différentielles. Souvent une étude de mouvement non périodique va s'exprimer par des fonctions hyperboliques. Tandis que la même étude de mouvement sans les forces de friction ou de résistance s'écrira avec des fonctions trigonométriques.

Les fonctions hyperboliques sont pré-définies dans *Mathematica*. On peut les dériver, les intégrer, les tracer, etc. Bref, les opérations vues dans les chapitres précédents sont également valables dans ce chapitre. Les quelques exemples classiques qui suivent illustrent bien ce fait.

Exemple 1.1.1.

Tracer la fonction sinus hyperbolique

```
Plot[Sinh[x], {x, -4, 4}]
```




```
-Graphics-
```

Exemple 1.1.2.

Dériver la fonction cosinus hyperbolique:

```
D[Cosh[x],{x,1}]
Sinh[x]
```

Exemple 1.1.3.

Montrer que $\sinh^2 x - \cosh^2 x = 1$

```
Simplify[Cosh[x]^2-Sinh[x]^2]
1
```

Exemple 1.1.4.

Trouver la dérivée seconde de $f(x) = \tanh(\sinh(x^2 + x + 1))$ pour $x = 0.2$.

```
D[Tanh[Sinh[x^2 + x + 1]],{x,2}]/.x->0.2
-0.898212
```

Exemple 1.1.5.

Calculer la primitive de $f(x) = 2x\,\mathrm{cosech}(x^2 + 8)$.

```
Integrate[2 x Csch[x^2 + 8],x]
```

$$-\text{Log}\left[\text{Cosh}\left[\frac{8 + x^2}{2}\right]\right] + \text{Log}\left[\text{Sinh}\left[\frac{8 + x^2}{2}\right]\right]$$

Exemple 1.1.6.

Trouver la limite de $\sinh x / x$ quand x tend vers zéro.

```
Limit[Sinh[x]/x,x->0]
1
```

Exemple 1.1.7.

Développer en série de MacLAURIN les fonctions: f(x) = cos x, g(x) = sin x et h(x) = $e^{i\,x}$.

Montrer que $e^{i\,x} = \cos x + i \sin x$.

Solution:

Soit la fonction **ns[fontion, x]** qui transforme **fonction** en série de MacLAURIN jusqu'à **x** élevé à la puissance neuf.

```
ns[fonction_,x_]:= Normal[Series[fonction,{x,0,9}]]
```

```
ns[Exp[I x],x]
```

$$1 + I\,x - \frac{x^2}{2} - \frac{I}{6}\,x^3 + \frac{x^4}{24} + \frac{I}{120}\,x^5 - \frac{x^6}{720} - \frac{I}{5040}\,x^7 + \frac{x^8}{40320} + \frac{I}{362880}\,x^9$$

```
Expand[ns[Cos[x],x] + I ns[Sin[x],x]]
```

$$1 + I\,x - \frac{x^2}{2} - \frac{I}{6}\,x^3 + \frac{x^4}{24} + \frac{I}{120}\,x^5 - \frac{x^6}{720} - \frac{I}{5040}\,x^7 + \frac{x^8}{40320} + \frac{I}{362880}\,x^9$$

Les deux expressions sont égales pour neuf termes, mais on aurait pu prendre n'importe quel nombre de termes.

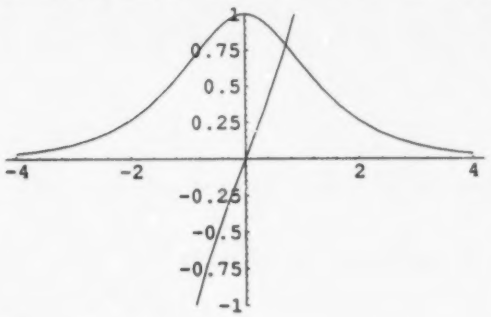
Exemple 1.1.8.

Trouver l'angle entre y = sinh x et sech x.

Solution:

Les graphes des deux fonctions sont:

```
Plot[{Sech[x],Sinh[x]},{x,-4,4},PlotRange->{-1,1}]
```




```
-Graphics-
```

La fonction **Solve[ ]** ne donne pas le point de rencontre directement nous devons donc l'aider en posant $y = e^{2x}$ dans les définitions de y = sech(x) et y = sinh(x). Ce problème survient lorsque la fonction **Solve[ ]** doit se servir des fonctions réciproques.

```
Solve[y^2 - 1 == 4 y,{y}]/.y->Exp[2x]
```

$$\{\{E^{2\,x} \to \frac{4 + 2\,\text{Sqrt}[5]}{2}\},\ \{E^{2\,x} \to \frac{4 - 2\,\text{Sqrt}[5]}{2}\}\}$$

Ensuite, on résout $y = e^{2x}$:

```
Solve[E^(2 x) == (4 + 2Sqrt[5])/2,{x}]

{{x -> Log[Sqrt[2 + Sqrt[5]]]}, {x -> Log[-Sqrt[2 + Sqrt[5]]]}}
```

On définit la fonction **theta[ ]** qui donne l'angle entre les tangentes au point où x est solution:

```
theta[_List]:= ArcTan[(m1-m2)/(1+m1 m2)];

liste={m1,m2}=
    {Evaluate[D[Sinh[x],x],D[Sech[x],x]]}/.x->
                                    Log[Sqrt[2 + Sqrt[5]]];
```

Finalement, le calcul de l'angle en degré:

```
theta[liste] 180/Pi//N

77.7409
```

Remarque.- Si on obtenait **ArcTan[ComplexInfinity]** comme réponse, il faudrait vérifier l'orthogonalité des courbes.

1.2 Les intégrales numériques.

Exemple 1.2.1.

On peut calculer la primitive d'une fonction et obtenir une réponse contenant des fonctions hyperboliques.

```
Integrate[Sqrt[x^2 + 4],x]
```

$$\frac{x\ \text{Sqrt}[4 + x^2]}{2} + 2\ \text{ArcSinh}[\frac{x}{2}]$$

Exemple 1.2.2.

On peut faire le calcul d'une intégrale définie, si la primitive existe:

```
Integrate[1/(4-x^2),{x,0,1}]
```

$$\frac{I}{4}\ \text{Pi} + \frac{-I\ \text{Pi} + \text{Log}[3]}{4}$$

```
%//N
```

```
0.274653 + 0. I
```

Exemple 1.2.3.

On peut faire le calcul d'une intégrale définie, même si la primitive n'existe pas:

```
NIntegrate[Sinh[x^2],{x,0,Pi}]
```

```
1633.13
```

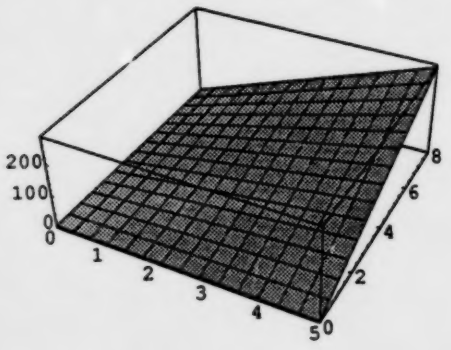
▫ Chapitre 2.

L'idée d'une fonction de plusieurs variables n'est pas nouvelle, lors de l'étude d'extrémum en calcul différentiel il nous est arrivé de travailler avec ce genre de fonctions. Cependant on avait toujours une condition qui nous permettait de faire la liaison entre les variables. La présente étude se fait avec des fonctions où les variables sont complètement indépendantes entre elles. Par exemple, le volume d'une boîte de hauteur constante h et de base rectangulaire est le produit de ses dimensions: $V(x, y) = x\ y\ h$. Les variables x et y n'ont aucun lien entre elles, elles sont indépendantes. Voyons un deuxième exemple, l'angle θ entre deux courbes qui s'intersectent en un point est une fonction de deux variables indépendantes m_1 et m_2. L'équation est: $\theta(m_1, m_2) = \arctan((m_1 - m_2) / (1 + m_1 m_2))$.

Voici les graphiques 3D des fonctions mentionnées plus haut:

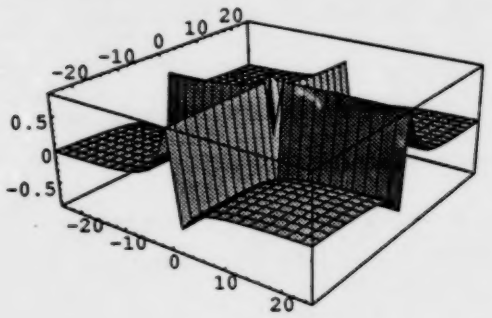
Le volume de la boîte dont les dimensions sont x, y et h; supposons que la hauteur de la boîte est $h = 7$ cm:

```
h = 7;
Plot3D[x y h ,{x,0,5},{y,0,8}]
```



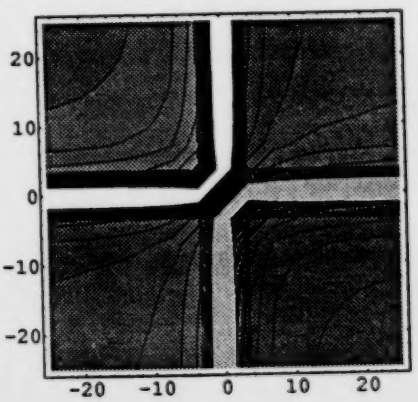
Le graphe de la deuxième fonction est plus surprenant, il donne une surface qui ondule dans l'espace:

```
Plot3D[ArcTan[(m1 - m2)/(1 + m1 m2)] ,{m1,-25,25},{m2,-25,25},
ViewPoint->{1.610, -2.110, 1.130},
PlotPoints->25]
```



Le domaine de définition de la fonction θ n'est pas R^2, puisque la fonction n'est pas définie à l'endroit où $m_1\ m_2 = -1$.

```
ContourPlot[ArcTan[(m1 - m2)/(1 + m1 m2)],
{m1,-25,25},{m2,-25,25},
Contours -> 20]
```



Ce dernier graphique donne les lignes de niveau de la surface définie par la fonction θ.

2.1 Les dérivées.

Les idées d'unicité d'image, de continuité, de dérivée, etc. lesquelles ont été discutées pour les fonctions d'une variable peuvent facilement être étendues à plusieurs variables. z = f(x, y) est une **fonction** de x et de y, si pour toutes valeurs x et y du domaine de définition de f, z est unique. La fonction f(x, y) est **continue** au point (a, b) si pour toute paire (x, y) dans le voisinage de (a, b), f(x, y) est dans le voisinage de f(a, b). La notion de dérivée pour une fonction de plusieurs variables prend différents sens selon les accroissements des variables.

La **dérivée** sera dite:

partielle, si une des variables prend un accroissement tandis que les autres sont gardées fixes;

totale, dy / dt, si f (x, y) est une fonction paramétrique, i.e. x = x(t) et y = y(t);

directionnelle, si elle (la dérivée) représente le taux de variation de **f** par rapport à **s** dans la direction θ, [**ds** =limite Δs =limite $\sqrt{(\Delta x^2 + \Delta y^2)}$ quand Δx et Δy tendent vers zéro].

Exemple 2.1.1.

Soit $z = x^2 y + x y^3$, trouver la dérivée partielle de z par rapport à x:

```
D[x^2 y + x y^3,x]
```

$2 x y + y^3$

Exemple 2.1.2.

La formule $a(x, y) = x^2 + x \sqrt{(x^2 + 4y^2)}$ donne l'aire de la surface d'une pyramide. Trouver les valeurs de ∂a /∂x et ∂a / ∂y au point x = 7 cm et x = 9 cm.

a) ∂a /∂x

```
a[x_,y_]:=x^2 + x Sqrt[x^2 + 4 y^2]

D[a[x,y],x]/.x->7/.y->9 //N
35.8503
```

Ce résultat signifie que l'aire de la surface de la pyramide croît au taux de 35.8503 cm^2 par cm pris sur x, quand y = 9 cm.

b) ∂a /∂y

```
D[a[x,y],y]/.x->7/.y->9 //N
13.0481
```

L'aire de la surface de la pyramide croît de 13.0481 cm^2 par cm dans le sens de l'ordonnée, quand l'abscisse est sur la droite x = 7 cm.

Exemple 2.1.3.

Un cube de minerai, dont les dimensions sont 10, 15 et 25 cm, a une densité en chaque point inversement proportionnelle au produit de ses coordonnées en ce point. Trouver le taux de variation de la densité au point (3, 5, 7) dans le sens de y quand x et z sont maintenus fixes.

Solution:

On cherche la dérivée partielle de la densité par rapport à y au point (3, 5, 7):

```
D[k/(x y z),y]/.{x->3,y->5,z->7}
```

$$\frac{-k}{525}$$

Ce taux de variation est inversement proportionnel à 525.

Exemple 2.1.4.

Trouver la dérivée totale de $z(x, y) = 5x^3 + 8x^2y^5 + \sqrt{(x\,y)}$ si $x(t) = \mathrm{Sin}(t^3)$ et $y(t) = e^{t^2}$

```
x[t_]:=Sin[t^3];
y[t_]:=Exp[t^2];
Dt[5 x[t]^3 + 8 x[t]^2 y[t]^5 + Sqrt[x[t] y[t]],t]
```

$$48\,E^{5t^2}\,t^2\,\mathrm{Cos}[t^3]\,\mathrm{Sin}[t^3] + 80\,E^{5t^2}\,t\,\mathrm{Sin}[t^3]^2 + 45\,t^2\,\mathrm{Cos}[t^3]\,\mathrm{Sin}[t^3]^2 + \frac{3\,E^{t^2}\,t^2\,\mathrm{Cos}[t^3] + 2\,E^{t^2}\,t\,\mathrm{Sin}[t^3]}{2\,\mathrm{Sqrt}[E^{t^2}\,\mathrm{Sin}[t^3]]}$$

Exemple 2.1.5.

Quel est le taux de variation du volume d'un cône circulaire droit sachant que:

la hauteur décroît à la vitesse de 20 cm / s quand elle est de 200 cm;

le rayon croît à la vitesse de 10 cm / s quand il est de 150 cm.

Solution:

```
Dt[Pi/3 r^2 h,t]/.Dt[r,t]->10/.Dt[h,t]->-20/.r->150/.h->200
50000 Pi
```

Exemple 2.1.6.

Trouver la dérivée directionnelle de $z = 5x^3 - 7x - 9xy^4 + 5y$ dans la direction $\theta = Pi/3$ au point (-1, 2, 156) de la surface.

Posons g = {D[z,x],D[z,y]} et u = {cos (t), sin(t) }, la dérivée directionnelle est le produit scalaire de g par u:

```
Clear[x,y,z,t];
z = 5 x^3 - 7 x -  9 x y^4 + 5 y;
g={D[z,x],D[z,y]};
theta= Pi/3;
u={Cos[theta],Sin[theta]};
deriveedirectionnelle = g.u /.{x->-1,y->2}
```

$$-68 + \frac{293\ \text{Sqrt}[3]}{2}$$

```
%//N
185.745
```

2.2 Le gradient.

Quand nous disons que la fonction est dérivée dans une direction θ nous voulons dire que le point $(x + \Delta x, y + \Delta y)$ s'approche du point (x, y) en restant sur une droite dont la pente est tan θ. En d'autres mots, Δx et Δy ne sont pas indépendants, ils vérifient l'équation $\tan\theta\,\Delta x = \Delta y$. En posant $\Delta s = \sqrt{(\Delta x^2 + \Delta x^2)}$, on peut trouver la limite de $\Delta f / \Delta s$, quand Δs tend vers zéro, c'est la dérivée directionnelle, notée $D_{(\theta)}f(x, y)$ ou df / ds. Si on représente respectivement par f_x et f_y, les dérivées partielles de f par rapport à x et à y, alors $D_{(\theta)}f(x, y) = f_x \cos\theta + f_y \sin\theta$. En un point donné, f_x et f_y sont des constantes, donc, $D_{(\theta)}f(x, y) = g(\theta)$, où g est une fonction de θ. Étant donné la nature de θ, le domaine de g, qui rend la fonction injective, est $D_g = [0, 2\pi[$.

```
Clear[fx, fy];
g[theta_]:=fx Cos[theta] + fy Sin[theta]
```

```
g'[theta]
fy Cos[theta] - fx Sin[theta]
```

La dérivée première g' est nulle, si sin (θ) / cos (θ) = f_y / f_x, i.e. tan(θ) = f_y / f_x.

```
g''[theta]
-(fx Cos[theta]) - fy Sin[theta]
```

Le signe de la dérivée seconde g" dépend de la valeur θ_σ qui résout l'équation trigonométrique tan(θ) = f_y / f_x. Si le signe de g"(θ_σ) est négatif, g(θ_σ) est un maximum et si le signe de g"(θ_σ) est positif, g(θ_σ) est un minimum. On peut démontrer que la valeur maximale est dans la direction du vecteur (f_x, f_y), et est égale au module du vecteur. Ce vecteur s'appelle le **gradient** de f.

La valeur minimale se trouve dans la direction opposée du vecteur **gradient**.

Exemple 2.2.1.

Considérons la fonction $z = 2x^3 - x y + 2y^3$. Du point (1, 2, 16) de la surface, quel est le taux de croissance maximum?

Solution:

Nous allons créer une fonction, appellée **modGrad[, ,]**, qui, dorénavent, fera le travail:

```
modGrad[z_,u_,v_]:=Block[{f,x,y,gv},
                         (grad[f_]:= {D[f,x],D[f,y]};
                         gv=grad[z]/.{x->u,y->v};
                         Sqrt[Plus @@ (gv^2)]//N)
                         ]
```

C'est une fonction à trois arguments: le premier représente la fonction, le deuxième l'abscisse du point et le troisième l'ordonnée du point.

```
modGrad[2 x^3 - x y + 2 y^3,1,2]
23.3452
```

Exemple 2.2.2.

Trouver la dérivée directionnelle maximale de z = cosh(x y) + sinh(x y) au point (2, -1, cosh(2) - sinh(2)).

Solution:

La dérivée est maximale dans la direction du gradient:

```
modGrad[Cosh[x y] + Sinh[x y],2,-1]
0.302619
```

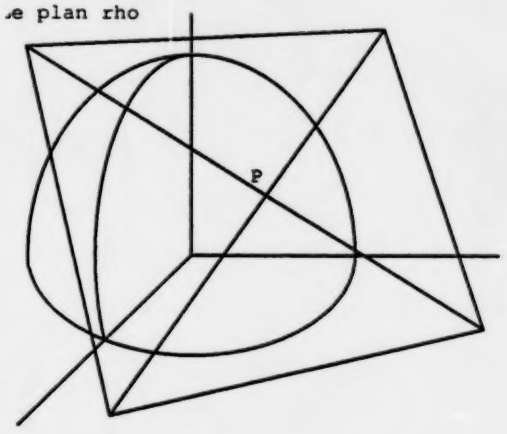
Quelle est l'interprétation graphique de la **dérivée directionnelle**? du **gradient**?

Une fonction z = f(x, y) forme dans l'espace 3D une surface. Les plans verticaux, i.e. parallèles à l'axe des z, découpent dans la surface des courbes. Le point P:(a, b, f(a, b)) appartient à la fois à la surface et à la courbe produite par l'intersection d'un certain plan vertical, disons π. De ce point P on peut tracer une tangente *l* à la courbe qui est entièrement contenue dans le plan π. Il va sans dire que la tangente *l* est aussi tangente à la surface. Toujours par le même point P on peut faire tourner le plan π, qui tracera dans la surface une autre courbe dont la tangente au point P:(a, b, f(a, b)) est *l**. Les deux tangentes *l* et *l** déterminent un plan ρ qui est tangent à la surface au point P:(a, b, f(a, b)). Le plan π qui a produit la tangente *l* fait un certain angle avec la partie positive de l'axe des x, disons θ. La **dérivée directionnelle** dans la direction θ au point P détermine la **pente** de la tangente *l*. Le plan qui a produit *l** fait un angle θ^* dont la **pente** de la tangente *l** est la **dérivée directionnelle** dans la direction θ^*. En bref, à chaque fois que le plan π tourne, tout en demeurant vertical, il produit une courbe qui a une tangente dont la pente est la dérivée directionnelle dans la direction de l'angle que le plan π fait avec la partie positive de l'axe des x. Le **module** du vecteur **gradient** donne la **plus grande de toutes les pentes** des tangentes qui passent par le point P dans le plan ρ.

La procédure suivante dessine une figure qui amène plus de clarté sur ce qui vient d'être dit:

```
ellipse[x_,y_,rx_,ry_,t1_,t2_]:= Circle[{x,y},{rx,ry},{t1,t2}]

axe[abs_,ord_]:=Line[{{0,0},{abs,ord}}]

Show[Graphics[{
     ellipse[0,0,4,5,0,3.2],ellipse[0,0,4,2.5,3.28,2Pi],
     ellipse[0,0,2.35,5,Pi/2,3.58],axe[6 Cos[5Pi/4],
     6 Sin[5Pi/4]],axe[0,6],axe[7.5,0],
```

```
Line[{{-4.04,5.21},{4.71,5.63},{7.13,-1.84},
{-2,-4},{-4.04,5.21}}],Line[{{-4.04,5.21},{7.13,-1.84}}],
Line[{{4.71,5.63},{-2,-4}}],
Text["P",{1.6,2}],Text[" Le plan rho",{-3,6}]
    }],
AspectRatio->Automatic]
```



Une famille de plans parallèles, perpendiculaires à l'axe des z, découpe la surface z = f(x, y) en une série de courbes appelées lignes de niveau. L'intersection d'un plan z = c et de la surface z = f(x, y) donne l'équation implicite f(x, y) = c. Cette équation représente une courbe dans le plan x0y. Soient R:(a+h, b+h) et Q:(a, b) deux points d'une même ligne de niveau f(x, y) = c, il est clair que f(R) - f(Q) = 0. La dérivée directionnelle de z = f(x, y) dans la direction de la tangente au point Q est: dz / ds =

gradient · **u** = $(f_x, f_y)\cdot(\cos\theta, \sin\theta) = (f_x, f_y)\cdot(-f_y/\sqrt{(f_x^2 + f_y^2)}, f_x/\sqrt{(f_x^2 + f_y^2)}) = 0$.

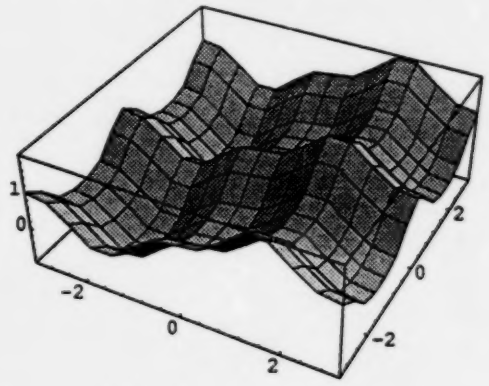
Puisque la projection vecteur **gradient** sur le vecteur direction **u** est nulle, il s'en suit que le **gradient** est perpendiculaire à la ligne de niveau au point (a, b).

Remarque.- Le **gradient** est un vecteur lié du plan x0y dont les composantes sont (f_x, f_y). Une ligne de niveau est une courbe définie par la fonction implicite f(x, y) = c dont le graphe est dans le plan x0y. Si on calcule les composantes du gradient au point (a, b) on arrive à la conclusion qu'il est perpendiculaire à la tangente de la ligne de niveau au point (a, b).

Exemple 2.2.3.

Soit la fonction $f(x, y) = \sin^3 x + \cos^4 x$ dont la représentation graphique est:

```
Plot3D[ Sin[x]^3 + Cos[y]^4,{x,-Pi,Pi},{y,-Pi,Pi}]
```



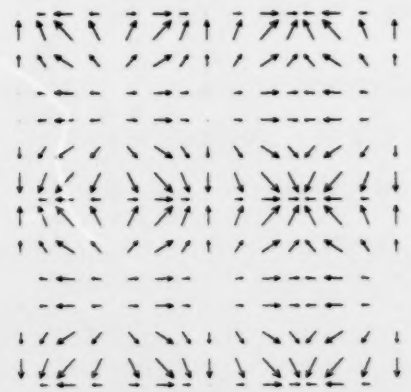
La surface ressemble à une chaîne de montagne.

Les deux graphiques qui suivent représentent respectivement un champ de gradients attachés aux points de la projection de la surface dans le plan horizontal et une carte de lignes de niveau.

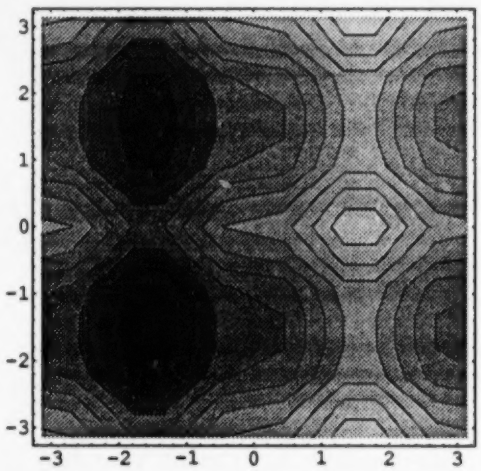
La fonction **PlotGradientField** est une fonction de *Mathematica* qui ne fait pas partie du bagage habituel de fonctions. Il faut ouvrir la filière par:

```
<<  Graphics`PlotField`
```

```
PlotGradientField[ Sin[x]^3 + Cos[y]^4,{x,-Pi,Pi},{y,-Pi,Pi}]
```



```
ContourPlot[ Sin[x]^3 + Cos[y]^4,{x,-Pi,Pi},{y,-Pi,Pi}]
```



Remarque.- Une façon de se représenter le gradient serait d'imaginer un skieur qui descend directement dans le sens de la plus grande pente «down-hill ou schuss». Il atteindra la vitesse maximale s'il est capable de se faire un réseau de lignes de niveaux imaginaires qui correspondent à la montagne et de s'y engager perpendiculairement. Cette ligne, qui colle à la pente, est le plus court chemin entre le sommet et l'arrivée. En d'autres mots, la courbe qui part du sommet et qui demeure tangente aux gradients en tout point a une projection dans le plan horizontal qui est perpendiculaire aux lignes de niveau.

En ce qui concerne une fonction de trois variables, soit $w = f(x, y, z)$ et

$\mathbf{u} = (\cos\alpha, \cos\beta, \cos\gamma)$ un vecteur unitaire, où α, β et γ sont les angles que le vecteur fait avec les axes de coordonnées.

On définit:

le **gradient** $= (f_x, f_y, f_z)$; où f_x, f_y et f_z sont les dérivées partielles de $w = f(x, y, z)$;

la **dérivée directionnelle = gradient** $\cdot$ **u**.

La dérivée directionnelle de $w = f(x, y, z)$ au point P est maximale dans la direction du gradient calculé en ce point.

Le gradient est perpendiculaire à la surface de niveau en ce point, i.e. au point (a, b, c, k) où $k = f(a, b, c)$. L'équation implicite de la surface de niveau est $k = f(x, y, z)$.

2.3 Les fonctions contraintes.

Qui n'a pas entendu parler du problème ancien: Marie, la rivière et la vache. La rivière sinueuse coule dans les champs. Sur le porche de la maison, Marie regarde au loin l'air songeur. Elle doit prendre un sceau près d'elle, aller à la rivière et abreuver la vache dans le pré voisin. Elle se demande: «Quel peut bien être le chemin le plus court pour aller de la maison à la rivière et de la rivière à la vache?»

La solution de ce problème consiste à tracer une famille d'ellipses dont les foyers sont respectivement M (Marie) et V (la vache). Les ellipses rayonnent des foyers et s'étendent dans le pré. La première ellipse à entrer en contact avec la rivière détermine le point P. Ce point P est tel que MP + PV est le chemin le plus court, par la définition même de l'ellipse.

C'est une solution à caractère géométrique, mais mathématiquement que se passe-t-il? Quand une des ellipses entre en contact avec la rivière le gradient de l'équation de l'ellipse et le gradient de l'équation de la rivière sont parallèles au point P. Au niveau des problèmes de ce type, l'équation de la rivière est considérée comme une contrainte en ce qui concerne l'équation de l'ellipse. Une contrainte dans le sens que s'il n'y avait pas de rivière le plus court chemin serait MV, i.e. de Marie à la vache.

Dans les exemples qui suivent nous allons nous servir de la notion de gradient pour résoudre des problèmes soumis à certaines contraintes.

Exemple 2.3.1.

Un plan passe par le point (a, b, c) du premier octant et détermine avec les plans de coordonnées un tétraèdre. Trouver les dimensions du tétraèdre afin que son volume soit minimum.

Nous allons donner deux solutions de ce problème: la première solution fait intervenir les dérivées partielles et la deuxième le paralllélisme des gradients.

La première solution:

Supposons que le plan coupe les axes de coordonnées aux points (u, 0, 0), (0, v, 0) et (0, 0, w), l'équation symétrique du plan dans le système cartésien est: x /u + y /v + z /w = 1. Comme le point (a, b, c) appartient au plan il doit vérifier son équation: a /u + b /v + c /w = 1, ceci nous donne une relation entre u, v et w. L'équation du volume du tétraèdre est V= 1 /6 u v w, or cette fonction n'est pas une fonction de trois variables indépendantes puisqu'il existe une relation entre u, v et w. Donc, la fonction volume peut se transformer pour devenir:

```
Clear[a,b,c,u,v,w,vol];
vol[u_,v_]:= 1/6 u v c u v /(u v - a v - b u);
```

```
Solve[{D[vol[u,v],u]==0,D[vol[u,v],v]==0},{u,v}]

{{u -> 0}, {u -> 3 a, v -> 3 b}, {u -> 0, v -> 3 b}, {u -> 0, v -> b},
  {v -> 0}}
```

Nous allons faire le calcul du déterminant de la matrice formée des dérivées partielles secondes afin de vérifier s'il existe bien un minimum.

```
Det[{{D[vol[u,v],u,u],D[vol[u,v],u,v]},
     {D[vol[u,v],v,u],D[vol[u,v],v,v]}}]/.{u->3 a,v->3 b}
```

$$\frac{3\ c^2}{4}$$

Pour u = 3a et v = 3b le déterminant est positif. Il existe effectivement un minimum qu'on vérifie par:

```
D[vol[u,v],u,u]/.{u->3 a,v->3 b}
```

$$\frac{b\ c}{a}$$

b c /a étant positif la concavité est donc vers le haut, d'où un minimum.

Pour calculer w à u = 3 a et v = 3b on se sert du lien entre les variables:

```
c u v /(u v - a v - b u)/.{u->3 a,v->3 b}

3 c
```

Donc, les dimensions du tétraèdre dont le volume est minimum sont 3a, 3b et 3c.

La deuxième solution:

Le gradient de la fonction volume est parallèle au gradient de la fonction contrainte.
La fonction volume est vol(u, v, w) = 1 /6 u v w et la fonction contrainte est donnée par l'équation du plan f(u ,v, w) = a /u + b/v + c /w (on remarquera que f(a, b, c) = 1, par l'équation symétrique du plan!):

```
Clear[a,b,c,u,v,w,f,vol];

vol[u_,v_,w_]:=1/6 u v w;

f[u_,v_,w_]:= a/u + b/v+ c/w;

gradientvol={D[vol[u,v,w],u],D[vol[u,v,w],v],D[vol[u,v,w],w]}
```

$$\{\frac{v\ w}{6},\ \frac{u\ w}{6},\ \frac{u\ v}{6}\}$$

```
gradientf={D[f[u,v,w],u],D[f[u,v,w],v],D[f[u,v,w],w]}
```

$$\{-(\frac{a}{u^2}),\ -(\frac{b}{v^2}),\ -(\frac{c}{w^2})\}$$

Les gradients sont parallèles, alors les composantes sont proportionnelles, disons à k, on déduit facilement les relations suivantes:

```
Solve[{a/u==b/v,L/v==c/w,c/w==a/u,a/u+b/v+c/w==1},{u,v,w}]
{{w -> 3 c, v -> 3 b, u -> 3 a}}
```

Donc, les dimensions du plus petit tetraèdre sont 3a, 3b et 3c.

Exemple 2.3.2.

Un coffret rectangulaire, joliment ciselé sur le dessus et les côtés, a un volume de 256 centimètres cubes. Quelles sont ses dimensions, si la surface ornée doit être minimale?

Solution:

La surface est soumise à la contrainte que le volume a 256 cm^3.
Les dimensions variables sont x, y et z.

```
Clear[vol,x,y,z];
vol[x_,y_,z_]:= x y z;
sur[x_,y_,z_]:= x y + 2 y z + 2 x z;
gradvol={D[vol[x,y,z],x],D[vol[x,y,z],y],D[vol[x,y,z],z]}
{y z, x z, x y}
gradsur={D[sur[x,y,z],x],D[sur[x,y,z],y],D[sur[x,y,z],z]}
{y + 2 z, x + 2 z, 2 x + 2 y}
Solve[{gradvol==k gradsur,x y z == 256},{x,y,z,k}]
{{k -> 2, z -> 4, y -> 8, x -> 8},
  {k -> 2 (-1)^(2/3), z -> 4 (-1)^(2/3), y -> 8 (-1)^(2/3), x -> 8 (-1)^(2/3)},
  {k -> 2 (-1)^(4/3), z -> 4 (-1)^(4/3), y -> 8 (-1)^(4/3), x -> 8 (-1)^(4/3)}}
```

Il s'en suit que les dimensions du coffret qui donnent la surface minimale sont 8, 8 et 4. Le coffret doit avoir une base carrée de 8 cm sur 8 cm et une hauteur de 4 cm.

Exemple 2.3.3.

Trouver le maximum et le minimum de $x^2 + y^2 + z^2$ sujet aux contraintes données par l'ellipsoïde e: $x^2 + y^2 + z^2/4 = 1$ et le plan p: $x + y + z = 0$.

Solution:

Comme nous avons deux contraintes, l'ensemble

{ gradient de $x^2 + y^2 + z^2$, gradient de e, gradient de p }

est linéairement dépendant.

Aussi, nous posons:

gradient ($x^2 + y^2 + z^2$) = k (gradient e) + m (gradient p).

```
d = x^2 + y^2 + z^2;
e = x^2 + y^2 + (z^2)/4 - 1;
p = x + y + z;

gradd={D[d,x],D[d,y],D[d,z]};
grade={D[e,x],D[e,y],D[e,z]};
gradp={D[p,x],D[p,y],D[p,z]};
Solve[{gradd == k grade + m gradp,e ==0,p==0},
{x,y,z,k,m}]
```

$$\{\{k \to 2,\ m \to \frac{-2}{\text{Sqrt}[3]},\ z \to \frac{-2}{\text{Sqrt}[3]},\ y \to \frac{1}{\text{Sqrt}[3]},\ x \to \frac{1}{\text{Sqrt}[3]}\},$$

$$\{k \to 2,\ m \to \frac{2}{\text{Sqrt}[3]},\ z \to \frac{2}{\text{Sqrt}[3]},\ y \to -(\frac{1}{\text{Sqrt}[3]}),\ x \to -(\frac{1}{\text{Sqrt}[3}$$

$$\{k \to 1,\ m \to 0,\ z \to 0,\ y \to -(\frac{1}{\text{Sqrt}[2]}),\ x \to \frac{1}{\text{Sqrt}[2]}\},$$

$$\{k \to 1,\ m \to 0,\ z \to 0,\ y \to \frac{1}{\text{Sqrt}[2]},\ x \to -(\frac{1}{\text{Sqrt}[2]})\}\}$$

Nous allons calculer les valeurs extrémales en placant les valeurs de x, y et z dans d:

```
d/.%
```

```
{2, 2, 1, 1}
```

Donc, la valeur maximale est 2 et la valeur minimale est 1. Ces nombres représentent la plus grande et la plus petite distance entre l'origine et l'intersection de l'ellipsoïde et du plan. Puisque cette courbe est une ellipse, le double de la racine carrée de ces nombres est respectivement la longueur du grand et du petit axe de l'ellipse.

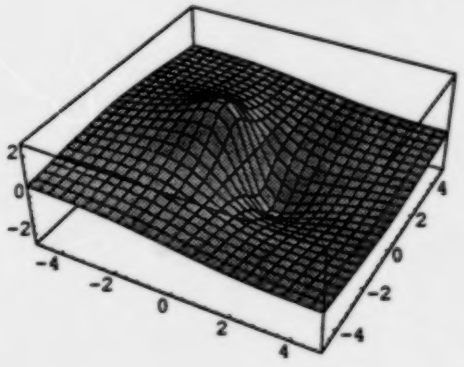
Exemple 2.3.4.

Trouver le maximum et le minimum de la fonction $z = -5\ x\ /(x^2 + y^2 + 1)$.

Solution:

D'abord le tracé de la surface:

```
Plot3D[- 5 x/(x^2 + y^2 +1),{x,-5,5},{y,-5,5},PlotPoints->25]
```



La figure parle d'elle-même. Le reste de la solution n'est que calcul de précision.

```
Clear[x,y];

z = (-5 x)/(x^2 + y^2 +1);

Solve[{D[z,x]==0,D[z,y]==0}]
```

```
{{x -> 1, y -> 0}, {x -> -1, y -> 0}}
```

Les points (1, 0) et (-1, 0) annulent les dérivées partielles. Ce sont les points critiques.

```
r={{D[z,x,x],D[z,x,y]},{D[z,y,x],D[z,y,y]}};

{Det[r]/.{x->1,y->0},Det[r]/.{x->-1,y->0}}
```

$$\left\{\frac{25}{4}, \frac{25}{4}\right\}$$

Le déterminant de la matrice des dérivées secondes aux points critiques est positif. Les points critiques sont donc des points d'extrémum et non pas des points de selle.

```
{D[z,x,x]/.{x->1,y->0},D[z,x,x]/.{x->-1,y->0}}
```

$$\{\frac{5}{2},\ -(\frac{5}{2})\}$$

D'après la concavité au point (1, 0) on a un minimum et au point (-1, 0) un maximum.

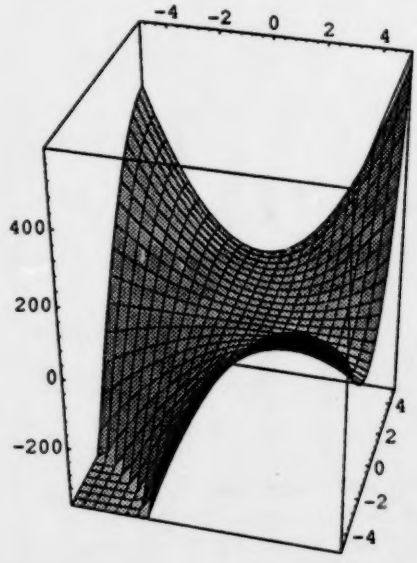
Exemple 2.3.5.

Trouver le minimum de la fonction $z = 4\,x^2 + 3\,x\,y^2 - 12\,x\,y$.

Solution:

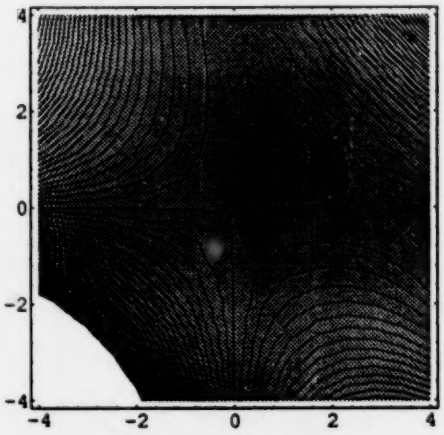
Nous allons faire un graphique pour la forme:

```
Clear[x,y,z];

Plot3D[4 x^2 y + 3 x y^2 - 12 x y,{x,-5,5},{y,-5,5},
PlotPoints->25,
ViewPoint->{0.920, -3.380, 2.310},
BoxRatios->{1,1,1.5}]
```



Il semble y avoir un point de selle et une dépression importante. Cernons cela de plus près avec la fonction qui trace les lignes de niveau.

```
ContourPlot[4 x^2 y + 3 x y^2 - 12 x y,{x,-4,4},{y,-4,4},
Contours->50]
```



Les lignes se regroupent selon deux tendances: une famille d'hyperboles équilatères et une famille d'ellipses concentriques. L'expérience montre que la famille d'hyperboles converge vers un point de selle et la famille d'ellipses vers un extrémum. Faisons les calculs afin de pousser plus loin l'expertise.

```
Clear[x,y,z];

z = 4 x^2 y + 3 x y^2 - 12 x y;

Solve[{D[z,x]==0,D[z,y]==0}]

{{x -> 1, y -> 4/3}, {x -> 0, y -> 4}, {x -> 3, y -> 0},
  {x -> 0, y -> 0}}

r=Det[{{D[z,x,x],D[z,x,y]},{D[z,y,x],D[z,y,y]}}];

{r/.{x->1,y->4/3},r/.{x->0,y->4},r/.{x->3,y->0},r/.{x->0,y->0}}

{48, -144, -144, -144}

D[z,y,y]/.{x->1,y->4/3}

6
```

Nous avons bien un minimum à (1, 4 /3), où la concavité est vers le haut. Au point (0, 0) les calculs ne révèlent pas grand chose les deux dérivées secondes s'annulent. Ce qui pourrait s'interpréter comme une sorte de plateau dans le voisinage de zéro. Cependant, en regardant le graphique 3D de la fonction z = f(x, y) il y a bien un point de selle à (0, 0).

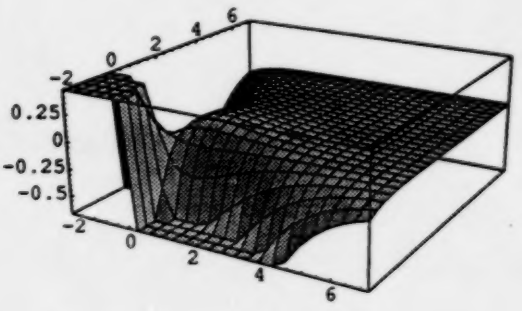
Exemple 2.3.6.

Soit la fonction $z = f(x, y) = x\, y\, e^{-x-y}$, trouver les points critiques. Donner une interprétation graphique à ces points.

Solution:

D'abord un graphique:

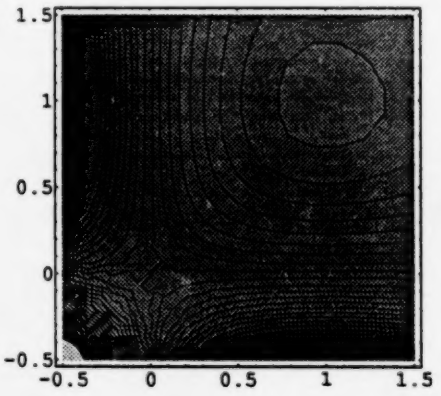
```
Plot3D[x y Exp[- x - y],{x,-2,7},{y,-2,7},PlotPoints->25,
ViewPoint->{1.610, -2.740, 1.130}]
```




```
-SurfaceGraphics-
```

Voici ce que nous révèle la fonction lignes de niveau:

```
Clear[x,y];
ContourPlot[x y Exp[- x - y],{x,-0.5,1.5},{y,-0.5,1.5},
Contours->50]
```



Il semble qu'il y aurait un point de selle à (0, 0) et un extrémum à (1, 1). Voyons si les calculs corroborent les interprétations graphiques.

```
Clear[x,y];
z = x y Exp[- x - y];
Simplify[{D[z,x],D[z,y]}/Exp[- x - y]]
```

```
{(1 - x) y, x (1 - y)}
```

```
Solve[{(1-x)y==0,x(1-y)==0},{x,y}]
```

```
{{x -> 1, y -> 1}, {x -> 0, y -> 0}}
```

```
r=Det[{{D[z,x,x],D[z,x,y]},{D[z,y,x],D[z,y,y]}}];
```

```
{r/.{x->0,y->0},r/.{x->1,y->1}}
```

```
{-1, E^-4}
```

```
D[z,y,y]/.{x->1,y->1}
```

```
-E^-2
```

Le point (0, 0) est un point de selle et le point (1, 1) un maximum relatif.

Exemple 2.3.7.

Trouver les constantes a et b pour lesquelles l'**intégrale définie** de la fonction $[\sin x - (a x^2 + b x)]^2$ entre 0 et π est un minimum.

```
Clear[a,b,x,f];

f[a_,b_]:=Integrate[(Sin[x]-(a x^2 + b x))^2,{x,0,Pi}];

Solve[{D[f[a,b],a]==0,D[f[a,b],b]==0},{a,b}]
```

$$\{\{a \to \frac{-60}{Pi^3} - \frac{4\ (80 - 20\ Pi^2)}{Pi^5},\ b \to \frac{48}{Pi^2} + \frac{3\ (80 - 20\ Pi^2)}{Pi^4}\}\}$$

```
%//N
{{a -> -0.400654, b -> 1.24798}}

r=Det[{{D[f[a,b],a,a],D[f[a,b],a,b]},
                {D[f[a,b],b,a],D[f[a,b],b,b]}}];

r/.%48
```

$$\{\frac{Pi^8}{60}\}$$

```
D[f[a,b],a,a]/.%48
```

$$\{\frac{2\ Pi^5}{5}\}$$

Quand a = - 0.400654 et b = 1.24798, les calculs ci-haut montrent bien que f(a, b) est un minimum. Le déterminant de la matrice des dérivées secondes est positif et la concavité est vers le haut, d'où minimum. Ce problème a été tiré du livre de Murray R. Spiegel *Theory and problems of ADVANCED CALCULUS* page 175, no 30. M. Spiegel ajoute que les idées invoquées dans ce problème sont importantes dans plusieurs branches des mathématiques et dans leurs applications. Il serait intéressant de s'attarder d'avantage sur ce type de problèmes, mais on doit passer à une autre partie du calcul qui traite des intégrales multiples.

▫ Chapitre 3.

3.1 Les Intégrales multiples.

Exemple 3.1.1.

Voici une série d'exemples sur les intégrales multiples que *Mathematica* exécute:

a) Une intégrale double avec la fonction exponentielle et la fonction cosinus hyperbolique:

```
Integrate[E^x + x + Cosh[y],{y,0,3},{x,0,2}]
```

$$3\ (1 + E^2) + 2\ \text{Sinh}[3]$$

```
%//N
```

```
45.2029
```

b) Une intégrale triple:

```
Integrate[4 x + 3 y + 2 z,{x,0,3},{y,0,2},{z,0,1}]
```

```
60
```

c) Une intégrale triple avec deux bornes variables:

```
Integrate[Integrate[Integrate[Cosh[x],{x,0,y+z}],{y,0,2 z}],{z,1,2}]
```

$$\text{Sinh}[1] - \text{Sinh}[2] - \frac{\text{Sinh}[3]}{3} + \frac{\text{Sinh}[6]}{3}$$

```
%//N
```

```
61.4468
```

d) Une intégrale triple où deux des bornes sont variables et la troisième paramétrique:

```
Integrate[Integrate[Integrate[x^2 + y^2 + z^2,{z,0,a-y-x}],
{y,0,a-x}],{x,0,a}]
```

$$\frac{a^5}{20}$$

Chapitre 3.

3.1 Les Intégrales multiples.

Exemple 3.1.1.

Voici une série d'exemples sur les intégrales multiples que *Mathematica* exécute:

a) Une intégrale double avec la fonction exponentielle et la fonction cosinus hyperbolique:

```
Integrate[E^x + x + Cosh[y],{y,0,3},{x,0,2}]
```

$$3\ (1 + E^2) + 2\ \text{Sinh}[3]$$

```
%//N
```

```
45.2029
```

b) Une intégrale triple:

```
Integrate[4 x + 3 y + 2 z,{x,0,3},{y,0,2},{z,0,1}]
```

```
60
```

c) Une intégrale triple avec deux bornes variables:

```
Integrate[Integrate[Integrate[Cosh[x],{x,0,y+z}],{y,0,2 z}],{z,1,2}]
```

$$\text{Sinh}[1] - \text{Sinh}[2] - \frac{\text{Sinh}[3]}{3} + \frac{\text{Sinh}[6]}{3}$$

```
%//N
```

```
61.4468
```

d) Une intégrale triple où deux des bornes sont variables et la troisième paramétrique:

```
Integrate[Integrate[Integrate[x^2 + y^2 + z^2,{z,0,a-y-x}],
{y,0,a-x}],{x,0,a}]
```

$$\frac{a^5}{20}$$

e) Une intégrale triple rencontrée dans le calcul des volumes:

```
Integrate[Integrate[Integrate[x y z,{z,Sqrt[x^2 + y^2],2}],
{y,0,1}],{x,0,1}]
```

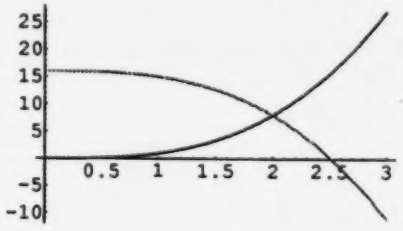
$$\frac{3}{8}$$

Les intégrales multiples servent à calculer les rayons de gyration d'une surface.

Exemple 3.1.2.

Trouver les rayons de gyration R_x et R_y de la surface plane limitée par $y = x^3$, $y = 16 - x^3$ et $x = 0$.

```
Plot[{x^3,16-x^3},{x,0,3},AspectRatio->1/GoldenRatio,
PlotStyle->{RGBColor[1,0,0],RGBColor[0,1,0]}]
```



On voit que

$x^3 < y < 16 - x^3$

$0 < x < 2$

```
Clear[x,y,d,rin];

in[d_]:=Integrate[Integrate[d^2,{y,x^3,16-x^3}],{x,0,2}]

rin={in[y],in[x]}/in[1]

Sqrt[rin]//N

{8.96979, 0.942809}
```

3.2 Les volumes.

Une façon d'interpréter le volume d'un solide est de considérer son contenu comme étant coincé entre deux surfaces. Un hémisphère, par exemple, est ce qui est entre une coquille sphérique et un disque ou d'une façon équivalente, sous une coquille sphèrique au-dessus d'un disque. En termes d'équations ce vocabulaire permet d'associer les variables aux relations qui déterminent les bornes d'intégration. Si on revient à l'hémisphère (de rayon r), les relations qui conditionnent les variables sont les suivantes:

z varie entre 0 et $\sqrt{(r^2 - x^2 - y^2)}$, l'équation de la coquille sphérique;

y varie entre 0 et $\sqrt{(r^2 - x^2)}$, l'équation du demi-cercle;

x varie entre 0 et r.

Elles délimitent les bornes d'intégration. Ce qui nous permet de faire le calcul d'un volume 3D à l'aide d'une intégrale double où normalement une intégrale triple serait requise.

Exemple 3.2.1.

Calculer le volume d'une sphère de rayon r.

Solution:

En tenant compte de ce qui a été dit plus haut , on écrit:

```
Integrate[Integrate[Sqrt[r^2 -x^2 -y^2],
                    {y,0,Sqrt[r^2 -x^2]}],
                        {x,0,r}]
```

$$\frac{Pi\ r^3}{6}$$

Ce calcul doit être multiplié par huit (*la surface est prise au-dessus d'un quart de disque*):

```
8 %
```

$$\frac{4\ Pi\ r^3}{3}$$

Ce résultat est bien le volume d'une sphère de rayon r.

Exemple 3.2.2.

Deux cylindres se coupent à angle droit, trouver le volume d'intersection.

L'intersection de deux cylindres peut être vue comme tout ce qui se trouve entre un tube horizontal et un disque. En d'autres mots, c'est ce qui est sous un tube et au-dessus d'un disque, le tout multiplié par deux afin de tenir compte de *la partie du haut et de la partie du bas*. On peut toujours imaginer les deux cylindres en forme de croix, l'un horizontal et l'autre vertical. Trouver le volume d'intersection consiste donc à calculer le volume sous une surface cylindrique au-dessus d'une région circulaire (*en coordonnées cylindriques les équations sont nettement plus simples*) et on trouve:

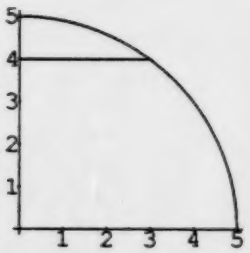
surface cylindrique	Sqrt[a^2 - rho^2 Cos[theta]^2]
région circulaire	rho = a
l'élément d'aire	rho d(theta) d(rho)

```
Integrate[Integrate[8 Sqrt[a^2 - rho^2 Cos[theta]^2] rho,
          {rho,0,a}],{theta,0, Pi/2}]
```

$$\frac{16\ a^2\ Sqrt[a^2]}{3}$$

C'est un type de problème où il est possible de faire une vérification avec une intégrale simple par la méthode des sections semblables (*les sections sont des parallélépipèdes à bases carrées dont le côté mesure*: $\sqrt{(a^2 - z^2)}$ *et la hauteur dz*).

Le volume de chaque section est $(a^2 - z^2)\,dz$. Une intégrale simple nous donne la somme entre $z = 0$ et $z = a$ de tous les volumes des sections carrées. Le résultat de cette intégrale doit être multiplié par huit, puisque le volume total se compose de huit octants.




```
Integrate[8 (a^2-z^2),{z,0,a}]
```

$$\frac{16\ a^3}{3}$$

Exemple 3.2.3.

a) L'ellipsoïde $b^2 x^2 + b^2 y^2 + a^2 z^2 = a^2 b^2$ est coupé par le cylindre $x^2 + y^2 = a x$, trouver le volume d'intersection;

b) Calculer le cas particulier où $a = 3$ et $b = 2$;

c) Trouver le volume d'intersection d'une sphère de rayon 2 a avec un cylindre de rayon a dont un des diamètres du cylindre coïncide avec un rayon de la sphère. Faire une vérification avec une intégrale simple en prenant $a = 5$.

Solution a)

Cas général

On peut imaginer le cylindre qui traverse horizontalement l'ellipsoïde. Trouver le volume d'intersection consiste à calculer le volume sous une surface cylindrique au-dessus d'une région elliptique.

Les équations sont traduites en coordonnées cylindriques.

Chaque équation devient:

cylindre $\quad rho^2 = a \cos[thêta]$

ellipsoïde $\quad b^2\ rho^2 + a^2\ z^2 = a^2\ b^2$

l'élément d'aire $\quad rho\ d(thêta)\ d(rho)$

```
Clear[a,b,v];
v=Integrate[Integrate[4 b/a Sqrt[a^2 -  rho^2] rho,
{rho,0,a Cos[theta]}],
{theta,0,Pi/2}]
```

$$\frac{-8\ a\ \text{Sqrt}[a^2]\ b}{9} + \frac{2\ a\ \text{Sqrt}[a^2]\ b\ \text{Pi}}{3}$$

Solution b)

Cas particulier

```
v/.{a->3,b->2}
-16 + 12 Pi
```

Remarque.- *Mathematica* attend de connaître la valeur intrinsèque de a avant de calculer **Sqrt[a^2]** qu'il traite temporairement comme | a | ou **Abs[a]**. Dans la solution générale a n'est pas connu, mais il n'en est pas de même dans la solution particulière, où a prend la valeur 3. À ce moment seulement $\sqrt{a^2} = a$ ou $\sqrt{3^2} = 3$, parce que a est positif. Si a avait été négatif nous aurions obtenu $\sqrt{a^2} = -a$.

Solution c)

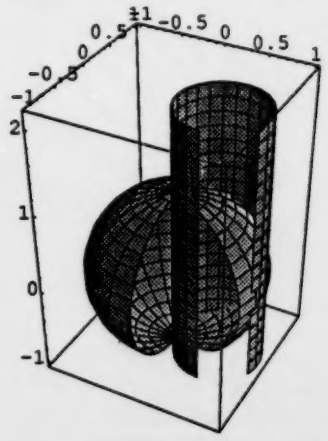
Cas général

L'équation de la sphère est: $z^2 = 4\,a^2 - x^2 - y^2$

L'équation du cylindre est: $y^2 = a^2 - (x - a)^2$.

La fonction **ParametricPlot3D[ ]** permet de dessiner la sphère et le cylindre. Les variables ne prennent pas tout l'espace qui leur est alloué afin de voir l'intérieur des figures.

```
ParametricPlot3D[{
     {Sin[v] Cos[u], Sin[v] Sin[u], Cos[v]},
     {1/2+Cos[u]/2, Sin[u]/2, v-1}
},
{u,-Pi/8,3Pi/2},{v,0,Pi}];
```



En terme de surface au-dessus d'une région, la surface est donnée par l'équation de la sphère et la région par le cylindre.

on voit que $0 < z < \sqrt{(4a^2 - x^2 - y^2)}$

$0 < y < \sqrt{[(a^2 - (x - a)^2]}$

$0 < x < 2a.$

D'où l'intégrale:

```
Integrate[Integrate[Sqrt[4 a^2 - x^2 - y^2],
                {y,0,Sqrt[a^2-(x-a)^2]}],{x,0,2 a}]
```

Or cette intégrale est très longue à calculer. On peut l'alléger par l'utilisation des coordonnées cylindriques.

En coordonnées cylindriques l'équation de la sphère est $z^2 = 4a^2 + rho^2$ et celle du cylindre est $rho = 2a\cos(thêta)$.

on voit que $0 < z < \sqrt{(4a^2 - rho^2)}$

$0 < y < 2a\cos(thêta)$

$0 < thêta < \pi/2$

d'où le volume est:

```
2 Integrate[Integrate[Sqrt[4 a^2 - rho^2] rho,
              {rho,0,2 a Cos[theta]}],{theta,0,Pi/2}]
```

$$2\left(\frac{-16\,a^2\,\mathrm{Sqrt}[a^2]}{9}+\frac{4\,a^2\,\mathrm{Sqrt}[a^2]\,\mathrm{Pi}}{3}\right)$$

```
%/.a->5
602.7531068
```

La vérification se fait par la méthode des sections semblables avec une intégrale simple.

Exemple 3.2.4.

Trouver le volume du solide borné par le cône $x^2 + y^2 = z^2$ et le paraboloïde $8\,z = x^2 + y^2 + 16$.

Solution:

En coordonnées cylindriques les équations sont de beaucoup simplifiées:

le cône $z = \mathrm{rho}$
le paraboloïde $z = 1/8\ \mathrm{rho}^2 + 2$
l'élément d'aire $dA = \mathrm{rho}\ d(\mathrm{thêta})\ d(\mathrm{rho})$

```
Solve[8 z == z^2 + 16,{z}]
{{z -> 4}, {z -> 4}}
```

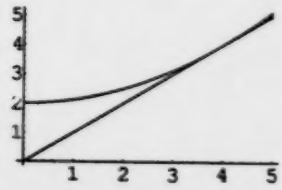
Le comportement des variables est le suivant:
z varie entre rho et 1/8 rho + 2, i.e. du cône au paraboloïde;
rho varie entre 0 et 4, le paraboloïde et le cône s'intersectent sur un cercle de rayon 4;
thêta varie entre 0 et 2π.

```
v = Integrate[Integrate[(1/8 rho^2 + 2 - rho) rho,
{rho,0,4}],
{theta,0,2 Pi}]
```

$$\frac{16\ Pi}{3}$$

Dans ce type de problème il est toujours possible de faire une vérification avec une intégrale simple par la méthode des couches cylindriques:

```
Plot[{x,(x^2 + 16)/8},{x,0,5}]
```




```
Integrate[2 Pi (1/8 x^2 + 2 - x),{x,0,4}]
```

$$\frac{16\ Pi}{3}$$

Exemple 3.2.5.

Trouver l'aire d'une sphère de rayon a dont l'équation cartésienne est $x^2 + y^2 + z^2 = a^2$.
Faire le problème en coordonnées cylindriques.

Solution:

Les transformations pour passer des coordonnées cartésiennes aux cylindriques sont:
z = Sqrt[a^2 - x^2 - y^2]
x=rho Cos[thêta]
y=rho Sin[thêta]
dA= rho d(rho)d(thêta)

On calcule l'aire d'un octant qu'on multiplie par huit afin d'obtenir l'aire de la sphère:

```
z = Sqrt[a^2 - x^2 - y^2];
x = rho Cos[theta];
y = rho Sin[theta];
```

```
Integrate[Integrate[8 Sqrt[1 + D[z,rho]^2 + D[z,theta]^2] rho,
          {rho,0,a}],{theta,0,Pi/2}]
```

$$4\,a^2\,\text{Pi}$$

Exemple 3.2.6.

Trouver l'aire de la face principale d'un tétraèdre formé par les plans de coordonnées et le plan π qui coupe les axes de coordonnées aux points a, b et c? (La face principale est celle qui est dans le plan π, les plans de coordonnées forment les faces secondaires).

Première solution: par intégrale

On a
$z = c(1-x/a-y/b)$, de l'équation symétrique du plan;
$0 < y < b(1-x/a)$, de l'équation symétrique de la droite qui passe par les points (a, 0) et (0, b);
$0 < x < a$.

```
Clear[x,y,z];

z = c(1-x/a-y/b);

aireprincipale=Integrate[Integrate[Sqrt[1 + D[z,x]^2 + D[z,y]^2],
                                   {y,0,b(1-x/a)}],{x,0,a}]
```

$$\frac{a\,b\,\text{Sqrt}\left[\frac{a^2\,b^2 + a^2\,c^2 + b^2\,c^2}{a^2\,b^2}\right]}{2}$$

Deuxième solution: par vecteur

Le produit vectoriel ne fait pas partie du bagage des fonctions lors de l'ouverture de *Mathematica*. Il faut charger la filière d'analyse vectorielle. Voici les commandes qu'il faut utiliser:

```
<<Calculus`VectorAnalysis`

SetCoordinates[Cartesian[x,y,z]]
```

Les vecteurs qui délimitent le triangle de la face principale du tétraèdre sont (-a, b, 0) et (-a, 0, c).

Nous allons définir le vecteur normal par le produit vectoriel de ces deux vecteurs:

```
normal = CrossProduct[{-a,b,0}, {-a,0,c}]
```

```
{b c, a c, a b}
```

L'aire du triangle est la moitié du module du vecteur normal:

```
aire2=1/2 Sqrt[Plus @@ (normal^2)]
```

$$\frac{\text{Sqrt}[a^2\,b^2 + a^2\,c^2 + b^2\,c^2]}{2}$$

Les réponses venant des deux solutions sont:

```
{aireprincipale,aire2}
```

$$\left\{\frac{a\,b\,\text{Sqrt}\left[\frac{a^2\,b^2 + a^2\,c^2 + b^2\,c^2}{a^2\,b^2}\right]}{2},\ \frac{\text{Sqrt}[a^2\,b^2 + a^2\,c^2 + b^2\,c^2]}{2}\right\}$$

Les deux réponses sont identiques. (Cf. voir la remarque 3.2.3).

Exemple 3.2.7.

Un solide est borné par le cylindre parabolique $z = y^2$ et les plans $x = 0$, $x = 2$ et $z = 1$. La fonction densité en tout point (x, y, z) est $z + 2$. Quelles sont les coordonnées du centre de gravité?

Définitions:

Le moment relatif au plan y0z est xm où m est la masse, i.e. la fonction de densité multipliée par l'élément de volume; noté: mx.

Le moment relatif au plan x0z est ym où m est la masse, i.e. la fonction de densité multipliée par l'élément de volume; noté: my.

Le moment relatif au plan x0y est zm où m est la masse, i.e. la fonction de densité multipliée par l'élément de volume; noté: mz.

L'abscisse du centre de gravité est définie comme étant le moment par rapport au plan y0z divisé par la masse totale.

L'ordonnée du centre de gravité est définie comme étant le moment par rapport au plan x0z divisé par la masse totale.

La cote du centre de gravité est définie comme étant le moment par rapport au plan x0y divisé par la masse totale.

La fonction cg[ ] (centre de gravité) définie plus bas représente une formule globale. Quand on remplace c par x, y, z et 1 on obtient respectivement mx, my, mz et le volume.
cgt[ ] donne les coordonnées du centre de gravité.

Solution (*exemple 3.2.7.*)

```
Clear[c,x,y,z,cg,cgt];

cg[c_]:=Integrate[Integrate[Integrate[c(z+2),
        {z,y^2,1}],{y,-1,1}],{x,0,2}];

cgt={cg[x],cg[y],cg[z]}/cg[1]
```

$$\left\{1,\ 0,\ \frac{57}{91}\right\}$$

Exemple 3.2.8.

Un solide est borné par les plans x = 0, x = 1, y = 0, z = 0 et par le paraboloïde $z = x^2 + y^2$. La fonction densité au point (x, y, z) est x y. Quelles sont les coordonnées du centre de gravité?

Solution:

```
Clear[c,x,y,z,cg,cgt];

cg[c_]:=Integrate[Integrate[Integrate[c(x y),
        {z,0,x^2+y^2}],{y,0,1}],{x,0,1}]

cgt={cg[x],cg[y],cg[z]}/cg[1]
```

$$\left\{\frac{11}{15},\ \frac{11}{15},\ \frac{7}{12}\right\}$$

Exemple 3.2.9.

Quel est le centre de gravité de la portion de la sphère qui est dans le premier octant?

Solution:

On suppose ici que la fonction densité prend la valeur un en tout point.

```
cg[c_]:=Integrate[Integrate[Integrate[c,
    {z,0,Sqrt[a^2-x^2-y^2]}],{y,0,Sqrt[a^2-x^2]}],{x,0,a}]
cgt={cg[x],cg[y],cg[z]}/cg[1]
```

$$\left\{\frac{3\ a}{8},\ \frac{3\ a}{8},\ \frac{3\ a}{8}\right\}$$

Exemple 3.2.10.

Un solide dont la projection dans le plan x0y est un quart de cercle de rayon a, celle dans le plan x0z est un rectangle de hauteur h et celle dans le plan y0z est un triangle isocèle. Ne considérer que la partie du solide située dans le premier octant et trouver le centre de gravité.

Solution:

Ici, le solide dont on parle est un conoïde.

On suppose que la fonction densité vaut un en tout point.

```
Clear[a,c,h,x,y,z,cg,cgt];
cg[c_]:=Integrate[Integrate[Integrate[c,
        {z,0,h(1-y/Sqrt[a^2-x^2])}],{y,0,Sqrt[a^2-x^2]}],
        {x,0,a}]
cgt={cg[x],cg[y],cg[z]}/cg[1]
```

$$\left\{\frac{2\ \text{Sqrt}[a^2]}{3\ \text{ArcSin}[\frac{a}{\text{Sqrt}[a^2]}]},\ \frac{4\ a}{9\ \text{ArcSin}[\frac{a}{\text{Sqrt}[a^2]}]},\ \frac{h}{3}\right\}$$

Exemple 3.2.11.

Quel est le centre de gravité d'un tétraèdre formé par les plans de coordonnées et le plan qui coupe les axes de coordonnées aux points a, b et c?

Solution:

La fonction densité prend la valeur **un** en tout point,

```
Clear[a,b,c,d,cg,cgt];

cg[d_]:=Integrate[Integrate[Integrate[d,
          {z,0,c(1- x/a- y/b)}],{y,0,b(1-x/a)}],{x,0,a}]

cgt = {cg[x],cg[y],cg[z]}/cg[1]
```

$$\{\frac{a}{4}, \frac{b}{4}, \frac{c}{4}\}$$

Exemple 3.2.12.

Deux sphères de rayon a sont imbriquées de telle sorte qu'une des sphères touche au centre de l'autre et vice versa. Calculer le volume de la partie à l'extérieur d'une des sphères.

Solution:

On place les sphères en position verticale, i.e. l'une sur l'autre.

Les coordonnées sphériques sont:
x = rho sin (phi) cos(thêta)
y = rho sin(phi) sin(thêta)
z = rho cos (phi)
dV = rho^2 sin(phi) d(phi) d(rho) d(thêta)

L'équation de la sphère de rayon a devient rho = a;

L'équation de la sphère de centre (0,0,a) est rho = 2 a sin(theta) =2 a cos(phi), où theta = Pi/2 - phi;

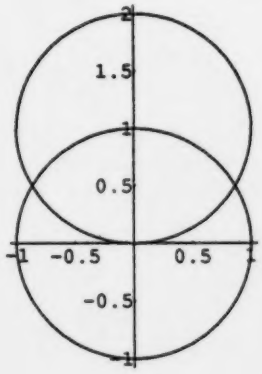
Les deux sphères se coupent sur le bord du cône phi = π/3

```
Clear[a];

v=Integrate[Integrate[Integrate[ 2 rho^2 Sin[phi],
                        {rho,a,2a Cos[phi]}],
                {phi,0,Pi/3}],
     {theta,0,2 Pi}]
```

$$\frac{11\ a^3\ Pi}{6}$$

Le graphique est:

```
a=1;
r[theta_]:= 2 a Sin[theta]

ParametricPlot[{{r[theta] Cos[theta],r[theta] Sin[theta]},
     {Cos[theta],Sin[theta]}},
     {theta,0,2 355/113},AspectRatio->Automatic]
```



Exemple 3.2.13.

Deux sphères de rayon a sont imbriquées de telle sorte qu'une des sphères touche le centre de l'autre et vice versa. Calculer le centre de gravité d'une des parties à l'extérieur de l'intersection.

Solution:

On utilise le graphique de l'exemple 3.2.12.

On place les sphères en position verticale.

Les coordonnées sphériques sont:
x = rho sin (phi) cos(thêta)
y = rho sin(phi) sin(thêta)
z = rho cos (phi)
dV = rho^2 sin(phi) d(phi) d(rho) d(thêta)

L'équation de la sphère de rayon a devient rho = a;

L'équation de la sphère de centre (0,0,a) est rho = 2 a cos(phi);

Les deux sphères se coupent sur le cône phi = Pi/3.

Quand nous remplaçons c par x=rho sin(phi)cos(thêta), y=rho sin(phi)sin(thêta), z=rho cos(phi) et 1 dans **cg[ ]**, nous obtenons respectivement mx, my, mz et le volume.

cgt[ ] donne les coordonnées du centre de gravité.

```
Clear[a,c,phi,rho,theta,cg,cgt,x,y,z];

x=rho Sin[phi]Cos[theta];
y=rho Sin[phi]Sin[theta];
z=rho Cos[phi];

cg[c_]:=Integrate[Integrate[Integrate[ c rho^2 Sin[phi],
              {rho,a,2a Cos[phi]}

                        ],{phi,0,Pi/3}
                            ],{theta,0,2 Pi}
              ];
cgt={cg[x],cg[y],cg[z]}/cg[1]
```

$$\left\{0,\ 0,\ \frac{27\ a}{22}\right\}$$

Chapitre 4.

4.1. Les équations différentielles.

Il existe dans *Mathematica* une pré-fonction qui permet de résoudre les équations différentielles. Voici quelques exemples qui illustrent ces possibilités du moins en ce qui concerne les équations de types classiques.

Exemple 4.1.1.

Les équations à variables séparables.

Résoudre $(1 + x^3)\, dy - x^2 y\, dx = 0$.

```
DSolve[y'[x]==(x^2  y[x])/(1+x^3),y[x],x]
```

$$\{\{y[x] \to (1 + x^3)^{1/3}\ C[1]\}\}$$

Exemple 4.1.2.

Les équations homogènes.

Résoudre $(x^2 - y^2)\, dx + 2\, x\, y\, dy = 0$.

```
DSolve[-(x^2-y[x]^2)/(2 x y[x]) ==y'[x],y[x],x]
{{y[x] -> Sqrt[x (-x + C[1])]}, {y[x] -> -Sqrt[x (-x + C[1])]}}
```

Exemple 4.1.3.

Les équations exactes.

Résoudre $(3\, x^2 + 1)\, y\, dx + (x^3 + x + 4\, y)\, dy = 0$

```
DSolve[(3 x^2 y[x] + y[x]) +(x^3 + x + 4 y[x])y'[x]==0,y[x],x]
```

$$\{\{y[x] \to \frac{-x - x^3 + \text{Sqrt}[x^2 + 2\, x^4 + x^6 + 8\, C[1]]}{4}\},$$

$$\{y[x] \to \frac{-(x + x^3 + \text{Sqrt}[x^2 + 2\, x^4 + x^6 + 8\, C[1]])}{4}\}\}$$

Exemple 4.1.4.

Les équations linéaires du premier ordre.

Résoudre $dy/dx + 3y/x = 8x^4$.

```
DSolve[y'[x]+3 y[x]/x == 8 x^4,y[x],x]
```

$$\{\{y[x] \to x^5 + \frac{C[1]}{x^3}\}\}$$

Exemple 4.1.5.

Les équations de Bernouilli.

```
DSolve[y'[x]+ 2 y[x]/x == x^3 y[x]^3,y[x],x]
```

$$\{\{y[x] \to \frac{1}{\text{Sqrt}[x^4\ (C[1] - 2\ \text{Log}[x])]}\},\ \{y[x] \to -(\frac{1}{\text{Sqrt}[x^4\ (C[1] - 2\ \text{Log}[x])]})\}\}$$

Dans ce qui suit nous donnons quelques exemples de problèmes où il faut former l'équation différentielle.

Exemple 4.1.6.

Si la population d'un pays triple en 60 ans, en combien d'années aura-t-elle quintuplé? (On suppose que le taux de croissance de la population est proportionnel au nombre d'habitants).

Solution

```
Clear[p,t,h];
DSolve[{p'[t] ==k p[t],p[0]==h}, p[t],t]
```

$$\{\{p[t] \to E^{k\ t}\ h\}\}$$

```
p[t_]:= E^(k t)
```

```
Solve[p[t]==5/3 p[60],{t}]/.k->Integrate[1/p,{p,1,3}]/60
```

$$\{\{t \to \frac{60\ \mathrm{Log}[5]}{\mathrm{Log}[3]}\}\}$$

```
%//N
```

```
{{t -> 87.8984}}
```

La population quintuplera en 87 ans et 11 mois.

Exemple 4.1.7.

Un corps placé dans un milieu qui a une température différente de celle de ce corps voit sa température varier à un taux qui est proportionnel à la différence entre la température du corps et celle du milieu ambiant.

Solution:

```
Clear[temperatureambiante,temperatureinitiale,
temperatureintermediaire,tempsintermediaire,tempsfinal];
DSolve[temperature'[temps]==k(temperature[temps] -
temperatureinitiale),temperature[temps],temps]
```

$$\{\{\mathrm{temperature}[\mathrm{temps}] \to \mathrm{temperatureinitiale} + E^{k\ \mathrm{temps}}\ C[1]\}\}$$

De cette équation différentielle on découvre la fonction **f[tempsfinal]** qui donne la température du corps après le temps final écoulé.

```
f[tempsfinal_]:= temperatureambiante +
(temperatureinitiale-temperatureambiante)*
((temperatureintermediaire-temperatureambiante)/
(temperatureinitiale-temperatureambiante))^(tempsfinal/
                                        tempsintermediaire)
```

Voici un cas particulier qui permet d'illustrer cette fonction.

Un corps à 100° est placé dans un milieu à 20°, si après 20 m le corps atteint une température de 70° quelle est la température de ce corps après 60 mimutes.

```
temperatureambiante = 20
temperatureinitiale = 100
temperatureintermediaire = 70
tempsintermediaire = 20;

20

100

70

f[60]//N

39.5312
```

Donc, un corps à 100° plongé dans un milieu à 20° qui atteint une température de 70° après 20 minutes aura une température de 39.53° une heure plus tard.

4.2 Les équations différentielles linéaires d'ordre élevé.

Exemple 4.2.1.

Une équation différentielle linéaire d'ordre deux, sans second membre.

Résoudre $d^2y/dx^2 + 5\ dy/dx + 4\ y = 0$.

```
DSolve[y''[x] + 5 y'[x] + 4 y[x] ==0,y[x],x]
```

$$\{\{y[x] \to \frac{C[1]}{E^{4\ x}} + \frac{C[2]}{E^{x}}\}\}$$

Exemple 4.2.2.

Une équation différentielle linéaire d'ordre deux, avec second membre.

Résoudre $d^2y/dx^2 + 5\ dy/dx + 4\ y = 6\ x - 4$.

```
DSolve[y''[x] + 5 y'[x] + 4 y[x] == 6 x - 4,y[x],x]
```

$$\{\{y[x] \to -(\frac{23}{8}) + \frac{3\ x}{2} + \frac{C[1]}{E^{4\ x}} + \frac{C[2]}{E^{x}}\}\}$$

Voici un exemple tiré de la physique.

Exemple 4.2.3.

Une chaîne de 4 m et de poids P est étirée sur une table dont le frottement est μ. On laisse pendre 1 m de chaîne par dessus le rebord de la table.

Solution:

i) Utiliser la relation **f = m a** pour montrer que pour toute longueur y ($y < 4$ m) dépassant la table, l'accélération est $d^2y/dx^2 = -g\mu + g(1+\mu)/4\ y$ où g est l'accélération terrestre;

ii) Trouver y après t secondes.

```
DSolve[{y''[t]== - g mu + g(1+mu)/4 y[t],y[0]==1,
y'[0]==0},y[t],t]
```

$$\{\{y[t] \to \frac{4\ mu}{1 + mu} + \frac{-3 + \frac{4}{1 + mu}}{2\ E^{(Sqrt[g]\ Sqrt[1 + mu]\ t)/2}} + \frac{E^{(Sqrt[g]\ Sqrt[1 + mu]\ t)/2}\ (-3 + \frac{4}{1 + mu})}{2}\}\}$$

Exemple 4.2.4.

Quelle est l'équation de la courbe qui traverse le point (0, 3) sous un angle de 135° et qui vérifie en tout point l'équation $d^2y/dx^2 = x + y$.

```
DSolve[{y''[x]== x +y[x],y[0]==3,y'[0]==-1},
    y[x],x]
```

$$\{\{y[x] \to \frac{3}{2\ E^x} + \frac{3\ E^x}{2} - x\}\}$$

Exemple 4.2.5.

Le bloc que l'on voit dans la figure ci-dessous a une masse de 160 /g Kg. Tel qu'on le voit, il est au repos entre deux ressorts sans tension sur une table lisse (i.e. sans frottement). Une force de 20 N par mètre est nécessaire pour étirer ou compresser l'un ou l'autre des ressorts.

i) Montrer que si le bloc est déplacé d'une distance x vers la droite, la force résultante agissante sur le bloc est de 40x N vers la gauche et son accélération est

$$d^2x/dt^2 = - g/4\ x.$$

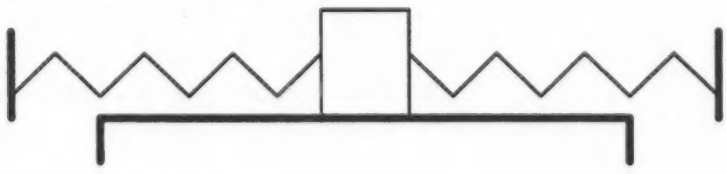
ii) Montrer que si le bloc est déplacé de 5 m vers la droite et ensuite relaché il se met à osciller par rapport à la position montrée et que l'équation du mouvement est

$$x = 5 \cos \sqrt{g}/2\ t$$

Solution:

Les fonctions ci-dessus tracent la figure de l'exemple 4.2.5.

```
bloc=Graphics[Line[{{1/2,0},{1/2,1.25},{-1/2,1.25},{-1/2,0}}]];
horiz=Graphics[{Thickness[.01],
              Line[{{-3,-1/2},{-3,0},{3,0},{3,-1/2}}]}];
verg=Graphics[{Thickness[.01],Line[{{-4,0},{-4,1}}]}];
verd=Graphics[{Thickness[.01],Line[{{4,0},{4,1}}]}];
lb=Graphics[Line[{{-4,.25},{-3.5,.75},{-3,.25},{-2.5,.75},{-2,.25},
{-1.5,.75},{-1,.25},{-.5,.75}}]];
lbd=Graphics[Line[{{4,.25},{3.5,.75},{3,.25},{2.5,.75},{2,.25},
{1.5,.75},{1,.25},{.5,.75}}]];
Show[{bloc,horiz,verg,verd,lb,lbd}
,AspectRatio->Automatic]
```



```
Clear[x,t,g];
DSolve[{x''[t]== - g/4 x[t] ,x[0]==5,x'[0]==0},
    x[t],t]
```

$$\{\{x[t] \to \frac{5\ E^{-I/2\ Sqrt[g]\ t}\ (1 + E^{I\ Sqrt[g]\ t})}{2}\}\}$$

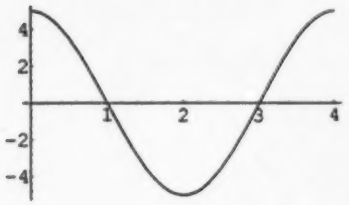
Si on utilise les lois de transformations trigonométriques:
E^(-I Sqrt[g]/2 t) -> Cos[Sqrt[g]/2 t] - I Sin[Sqrt[g]/2 t];
E^(I Sqrt[g]/2 t) -> Cos[Sqrt[g]/2 t] + I Sin[Sqrt[g]/2 t].

La simplification donne une fonction réelle qui peut être facilement tracée.

```
Simplify[5(Cos[Sqrt[g]/2 t]- I Sin[Sqrt[g]/2 t])(1 +
(Cos[Sqrt[g] t]+ I Sin[Sqrt[g] t]))]/2
```

$$5\ \text{Cos}\left[\frac{\text{Sqrt}[g]\ t}{2}\right]$$

```
Plot[5 Cos[Sqrt[9.8]/2 t],{t,0,4}]
```



Exemple 4.2.6.

Si on place le bloc de l'exemple 4.2.5. dans un milieu où la force de résistance est proportionnelle à la vitesse, montrer que l'équation différentielle est

$$160/g\ d^2x/dx^2 + k\ dx/dt + 40\ x = 0.$$

où k est une constante positive.

```
Clear[x,t,g];
DSolve[{160/g x''[t] + k x'[t]+ 40 x[t] == 0 ,x[0]==5,x'[0]==0},
    x[t],t]
```

$$\{\{x[t] \to \left(E^{((-(g\ k) - \text{Sqrt}[g]\ \text{Sqrt}[-25600 + g\ k^2])\ t)/320}\right.$$
$$\left.\left(5 - \frac{5\ \text{Sqrt}[g]\ k}{\text{Sqrt}[-25600 + g\ k^2]}\right)\right) / 2 +$$
$$\left(E^{((-(g\ k) + \text{Sqrt}[g]\ \text{Sqrt}[-25600 + g\ k^2])\ t)/320}\right.$$
$$\left.\left(5 + \frac{5\ \text{Sqrt}[g]\ k}{\text{Sqrt}[-25600 + g\ k^2]}\right)\right) / 2\}\}$$

Cas particulier où k = 10

```
%4/.{k->10,g->9.8}
```

$$\{\{x[t] \to (2.5 + 0.49878\ I)\ E^{(-0.30625 - 1.535\ I)\ t} +$$
$$(2.5 - 0.49878\ I)\ E^{(-0.30625 + 1.535\ I)\ t}\}\}$$

Cas particulier où k = 100

```
%4/.{g->9.8,k->52}
```

$$\{\{x[t] \to \frac{-11.0715}{E^{1.88585\ t}} + \frac{16.0715}{E^{1.29915\ t}}\}\}$$

Le mouvement sera oscillatoire seulement pour les valeurs de k < Sqrt[25600 g].

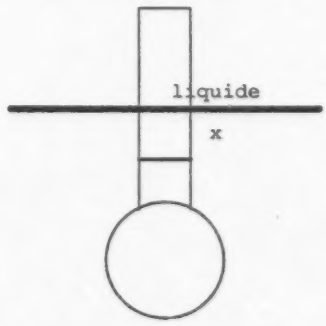
Exemple 4.2.7.

Un densimètre est enfoncé d'une profondeur de x cm dans un liquide et ensuite il est relâché afin d'étudier son mouvement. La section de la tige du densimètre mesure A cm^2, sa masse est m, la densité du liquide est ρ et la constante de friction fluide est β. Donner l'équation du mouvement.

Solution:

Les fonctions qui suivent permettent de tracer le graphique du densimètre.

```
horiz=Graphics[{Thickness[.02],
              Line[{{-3,0},{3,0}}]}];

tige=Graphics[Line[{{-1/2,-2},{-1/2,2},{-1/2,2},
{1/2,2},{1/2,2},{1/2,-2}}]];

reticule=Graphics[{Thickness[.01],
              Line[{{-1/2,-1},{1/2,-1}}]}];

hb=Graphics[{Thickness[.005],Circle[{0,-3},1.15]}];
t=Graphics[Text["x ",{1,-1/2}]];
tt=Graphics[Text["     liquide ",{1,1/3}]];
Show[{horiz,tige,hb,reticule,t,tt},

AspectRatio->Automatic]
```



Le densimètre a une certaine flottabilité, mais lorsqu'il est enfoncé dans le liquide il reçoit une poussée additionnelle $f_a = (\rho\ g\ A)\ x$, dite poussée d'Archimède. La force qui s'oppose au mouvement est la friction fluide $f_\beta = -\ \beta\ dx/dt$.

Puisque **f** = m **a** on trouve donc

$$(\rho\ g\ A)\ x - \beta\ dx/dt = m\ d^2x/dt^2$$

D'où l'équation différentielle

$$d^2x/dt^2 - \beta/m\ dx/dt + (\rho\ g\ A)\ x/m = 0$$

```
Clear[x,t,g,beta,rho, A];
DSolve[x''[t] + beta/m x'[t]+ rho g A /m x[t] == 0,x[t],t]
```

$$\{\{x[t] \to \frac{C[1]}{E^{((beta + Sqrt[beta^2 - 4\ A\ g\ m\ rho])\ t)/(2\ m)}} + E^{((-beta + Sqrt[beta^2 - 4\ A\ g\ m\ rho])\ t)/(2\ m)}\ C[2]\}\}$$

Conditions particulières.

Supposons que:
le densimètre est enfoncé de 1 cm;
la masse est 1/4 Kg;
la section de la tige mesure 3/4 cm^2
la densité est 1.5;
la friction fluide est 2.5

```
Clear[x,t,g,beta,rho,A];

DSolve[{x''[t] + beta/m x'[t]+ rho g A /m x[t] == 0,
x'[0]==0,x[0]==-1},x[t],t]
```

$$\{\{x[t] \to \frac{E^{((-beta + Sqrt[beta^2 - 4\ A\ g\ m\ rho])\ t)/(2\ m)}\ (-1 - \frac{beta}{Sqrt[beta^2 - 4\ A\ g\ m\ rho]}))\ /\ 2 + \frac{-1 + \frac{beta}{Sqrt[beta^2 - 4\ A\ g\ m\ rho]}}{2}}{E^{((beta + Sqrt[beta^2 - 4\ A\ g\ m\ rho])\ t)/(2\ m)}}\}\}$$

Avec les valeurs particulières on trouve

```
%2/.{beta->25,
A->3/4,m->250,rho->1.5,g->9.8}
```

$$\{\{x[t] \to (-0.5 - 0.122573\,I)\,E^{(-0.05 - 0.203961\,I)\,t} + (-0.5 + 0.122573\,I)\,E^{(-0.05 + 0.203961\,I)\,t}\}\}$$

Si la friction liquide est relativement faible nous obtenons un mouvement oscillatoire amorti.

La valeur de β ne peut pas être aussi grande que l'on veut. Pour des valeurs de β plus grandes que le calcul ci-dessous les équations intermédiaires font intervenir les nombres complexes.

```
Sqrt[4 3/4 9.8 250 1.5]
105.
```

Si la friction liquide est plus grande que 105, en tenant compte des conditions particulières, le mouvement n'est plus oscillatoire, la viscosité est trop grande et il y a changement de mouvement. Le densimètre monte vers la surface du niveau du liquide. Il s'arrête à la surface sans la dépasser. Selon la viscosité le temps que le densimètre prend avant de s'arrêter sera plus ou moins long.

```
%2/.{beta->110,
A->3/4,m->250,rho->1.5,g->9.8}
```

$$\{\{x[t] \to \frac{1.17748}{E^{0.285574\,t}} - \frac{2.17748}{E^{0.154426\,t}}\}\}$$

Revenons aux premières données où β était égal à 25. Si on utilise les lois de transformations trigonométriques:

$$E^{-I\,b\,t} \to \text{Cos}\,b\,t - I\,\text{Sin}\,b\,t;$$
$$E^{I\,b\,t} \to \text{Cos}\,b\,t + I\,\text{Sin}\,b\,t.$$

La simplification donne une fonction réelle qui peut être facilement tracée.

$$\{\{x[t] \to (-0.5 - 0.122573\,I)\,E^{(-0.05 - 0.203961\,I)\,t} + (-0.5 + 0.122573\,I)\,E^{(-0.05 + 0.203961\,I)\,t}\}\}$$

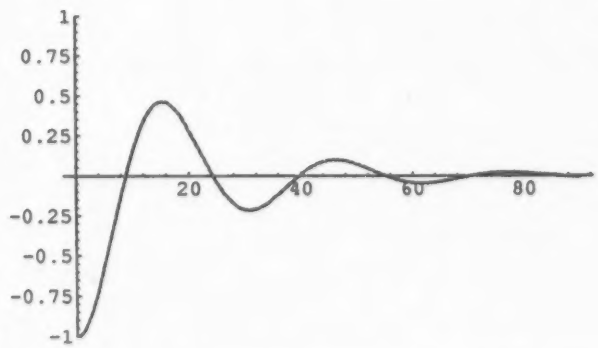
```
Simplify[(-0.5 - 0.122573 I)
(
E^(-0.05 t)(Cos[0.203961 t]-I Sin[0.203961 t])
)+
(-0.5 + 0.122573 I)
(
E^(-0.05 t)(Cos[0.203961 t]+I Sin[0.203961 t]))
]
```

$$\frac{-1.\ \text{Cos}[0.203961\ t]}{E^{0.05\ t}} - \frac{0.245146\ \text{Sin}[0.203961\ t]}{E^{0.05\ t}}$$

Du résultat, formons et traçons la fonction x[t]:

$$x[t_] := \frac{-1.\ \text{Cos}[0.203961\ t]}{E^{0.05\ t}} - \frac{0.245146\ \text{Sin}[0.203961\ t]}{E^{0.05\ t}}$$

```
Plot[x[t],{t,0,90},
PlotRange->{-1,1},
PlotStyle->RGBColor[1,0,0]]
```



La figure ci-dessus montre bien un mouvement oscillatoire amorti.

Chapitre 5.

5.1 Les séries de Fourier.

En plus des séries de puissances il y a une autre classe de séries infinies qui jouent un rôle important en mathématiques pures et appliquées. Ce sont les séries de Fourier dont les termes individuels sont des fonctions trigonométriques et la somme de ces termes est une fonction périodique.

Une fonction périodique de période **p** est représentée par l'équation $f(x + p) = f(x)$ pour toutes les valeurs de **x**.

Si on pose $x = t$, où t représente le temps, la fonction $f(t)$ s'interprète comme une vibration ou une oscillation. La période **p** est appelée la période de vibration ou d'oscillation.

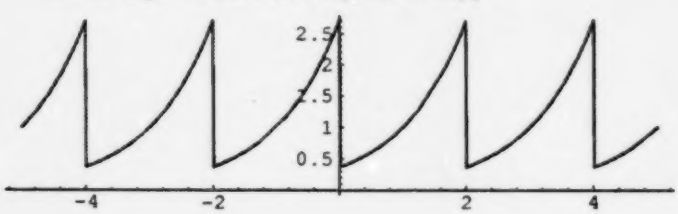
Une fonction définie sur l'intervalle $-p/2 \leq x \leq p/2$ peut toujours être étendue à l'ensemble des nombres réels afin de devenir une fonction périodique, si en dehors de l'intervalle, nous définissons $f(x)$ par $f(x + n\,p/2)$, où n est un entier quelconque.

Voici un fait intéressant concernant les fonctions périodiques: l'intégrale d'une fonction périodique sur un intervalle de longueur égale à la période **p** a toujours la même valeur peu importe où se trouve l'intervalle.

Exemple 5.1.1.

Voici le tracé d'une fonction périodique de période 2. La fonction **Table[ ]** donne une suite d'intégrales dont l'intervalle d'intégration est 2 pour toutes les intégrales. Le balayage part de -3 pour se rendre jusqu'à 3. Il y a en tout 33 calculs effectués. Tous ces calculs sont égaux, ce qui illustre bien ce qui a été dit plus haut.

```
Clear[f];

f[x_]:=Mod[x,2]-1

Plot[Exp[f[x]],{x,-5,5},AspectRatio->Automatic,
     PlotStyle->RGBColor[.5,0,.5]]
```



```
Table[NIntegrate[Exp[f[x]],{x,-1+i,1+i},WorkingPrecision -> 9],
      {i,-2,2,.125}]
```

```
{2.350402387287602914, 2.350402387287602914, 2.350402387287602914,
 2.350402387287602914, 2.350402387287602914, 2.350402387287602914,
 2.350402387287602914, 2.350402387287602914, 2.350402387287602914,
 2.350402387287602914, 2.350402387287602914, 2.350402387287602914,
 2.350402387287602914, 2.350402387287602914, 2.350402387287602914,
 2.350402387287602914, 2.350402387287602914, 2.350402387287602914,
 2.350402387287602914, 2.350402387287602914, 2.350402387287602914,
 2.350402387287602914, 2.350402387287602914, 2.350402387287602914,
 2.350402387287602914, 2.350402387287602914, 2.350402387287602914}
```

Avec les séries de Fourier nous allons échafauder des fonctions périodiques très générales à partir des fonctions **a sin ωx** et **a cos ωx**.

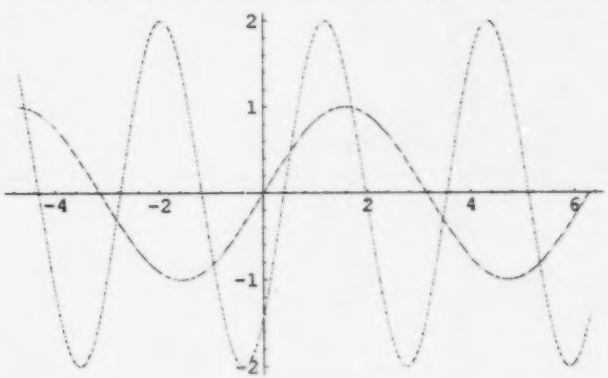
Les vibrations sinusoïdales ou les vibrations harmoniques simples sont représentées par les fonctions **a sin ω(x - ξ)** et **a cos ω(x - ξ)**.

La période de vibration est $T = 2\pi / \omega$;
la fréquence est $1/T$;
l'amplitude est a;
la phase est le nombre ξ dans $\omega(x - \xi)$.

Exemple 5.1.2.

Dans le graphique qui suit l'amplitude est 2, la période est π et la phase est $\pi/8$.

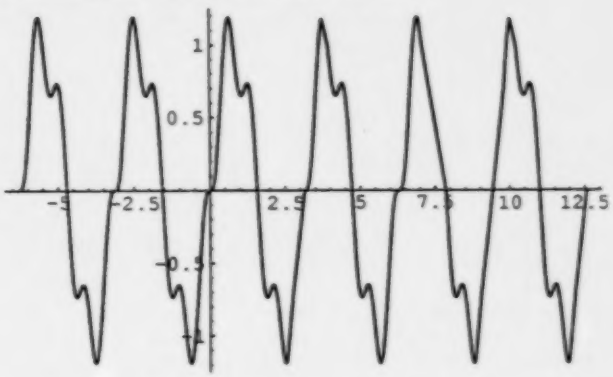
```
Plot[{Sin[x],2 Sin[2(x - Pi/8)]},{x,-3 Pi/2,2 Pi},
PlotStyle ->{RGBColor[1,0,.5],RGBColor[.2,.5,1]}]
```



Exemple 5.1.3.

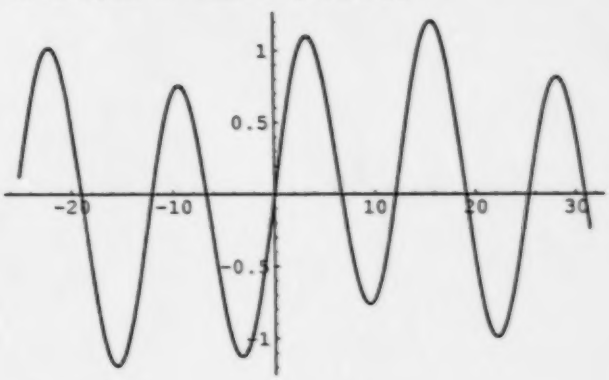
La somme de deux fonctions périodiques donne une troisième fonction périodique.
La période de la fonction y = sin 2x est π et la période de la fonction 1/4 sin 8(x -Pi/8) est π /4. La période de la fonction somme est π. En effet, le graphique qui suit nous montre bien cette affirmation.

```
Plot[Sin[2x]+.25 Sin[8(x - Pi/8)],{x,-2 Pi,4 Pi},
PlotStyle->RGBColor[0,0,1]]
```



Essayons avec deux fonctions dont les périodes sont respectivement 4 π et 14 π:

```
Plot[Sin[1/2 x]+.25 Sin[1/7(x - Pi/8)],{x,-8 Pi,10 Pi},
PlotStyle->RGBColor[0,0,1]]
```



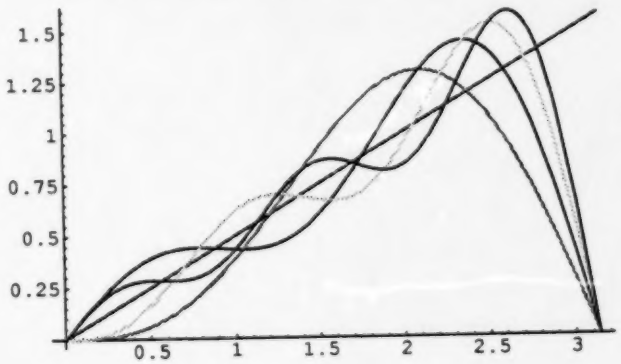

```
-Graphics-
```

Exemple 5.1.4.

Nous allons maintenant utiliser des fonctions dont la forme est f(x) = 1/n sin(n x), où n est un entier. Les périodes et les amplitudes sont respectivement 1 et π, 1/2 et π/2, 1/3 et π/3, 1/4 et π/4 et 1/5 et π/5. La superposition de la différence alternée de ces fonctions sur un même graphique donne un graphe qui tend vers le graphe de y = x/2 périodique.

```
s1[x_]:=Sin[x];
s2[x_]:=Sin[2 x]/2;
s3[x_]:=Sin[3 x]/3;
s4[x_]:=Sin[4 x]/4;
s5[x_]:=Sin[5 x]/5;
```

```
Plot[{x/2,
      s1[x]-s2[x],
      s1[x]-s2[x]+s3[x],
      s1[x]-s2[x]+s3[x]-s4[x],
      s1[x]-s2[x]+s3[x]-s4[x]+s5[x]},{x,0,Pi},
      PlotStyle->{RGBColor[1,0,0],RGBColor[0,1,0],
      RGBColor[0,0,1],RGBColor[1,1,0],RGBColor[.5,0,1]}]
```




```
-Graphics-
```

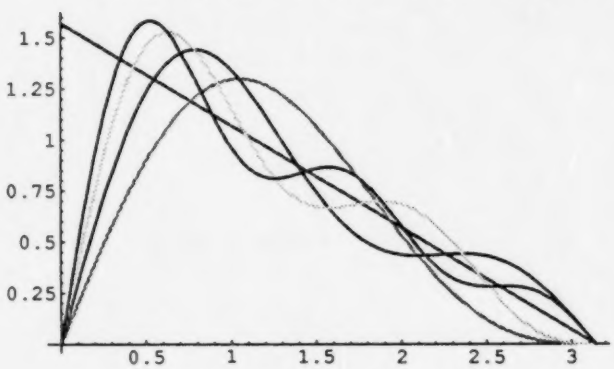
On écoute cette fonction avec la fonction **Play[ ]**.

```
Play[{x/2,
     s1[x]-s2[x],
     s1[x]-s2[x]+s3[x],
     s1[x]-s2[x]+s3[x]-s4[x],
     s1[x]-s2[x]+s3[x]-s4[x]+s5[x]},{x,0,Pi}]
```

Exemple 5.1.5.

Un phénomène semblable se produit avec la somme de ces fonctions.

```
Plot[{Pi/2-x/2,
     s1[x]+s2[x],
     s1[x]+s2[x]+s3[x],
     s1[x]+s2[x]+s3[x]+s4[x],
     s1[x]+s2[x]+s3[x]+s4[x]+s5[x]},{x,0,Pi},
     PlotStyle->{RGBColor[1,0,0],RGBColor[0,1,0],
     RGBColor[0,0,1],RGBColor[1,1,0],RGBColor[.5,0,1]}]
```



Exemple 5.1.6.

Pour une fonction donnée f, est-il possible de trouver un jeu de constantes de telle sorte que la somme de ces fonctions (Cf. 5.1.4.) peut approximer la fonction f sur l'intervalle $-\pi \leq x \leq \pi$? Plus précisément, la fonction f peut-elle être développée en une somme infinie de fonctions trigonométriques?

Cette somme infinie de fonctions est une série de Fourier.

Voici quelques théorèmes mathématiques qui répondent à cette question:

• Si la fonction est *lisse* par section et si $f(x) = 1/2(f(x - 0) + f(x + 0))$, alors le développement en série de Fourier converge en tout point et représente la fonction.

• Sur tout intervalle où la fonction est *continue* et *lisse* par section, la série de Fourier converge uniformément.

• Si la fonction est *lisse* par section et *continue*, la serie de Fourier converge uniformement.

Les séries de Fourier:
$s_n(x) = 1/2\ a_0 + \Sigma(a_j \cos j\ x + b_j \sin j\ x)$ où j varie de 1 à n,

$a_j = 1/\pi \int f(x) \sin j\ x\ dx$ entre $-\pi$ et $+\pi$;
$b_j = 1/\pi \int f(x) \cos j\ x\ dx$ entre $-\pi$ et $+\pi$.

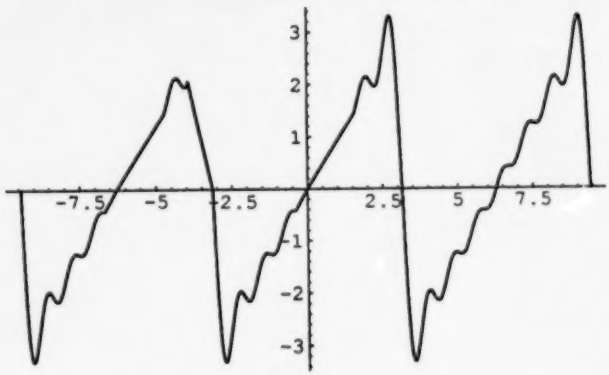
```
Clear[fa,b,j,n,f,x];
fa[j_]:=Integrate[f[x] Cos[j x],{x,-Pi,Pi}]/Pi;
fb[j_]:=Integrate[f[x] Sin[j x],{x,-Pi,Pi}]/Pi;
```

Exemple 5.1.7.

Développer en série de Fourier la fonction f(x) = x (sept termes). Tracer le graphique.

```
Clear[fa,fb,j,n,f,x,s];
fa[j_]:=Integrate[f[x] Cos[j x],{x,-Pi,Pi}]/Pi;
fb[j_]:=Integrate[f[x] Sin[j x],{x,-Pi,Pi}]/Pi;
```

```
f[x_]:= x;
s = Table[fa[j] Cos[j x]+ fb[j] Sin[j x],{j,1,7}];
AppendTo[s,fa[0]/2];
seriefourier[x ]:= Plus @@ s
Plot[seriefourier[x],{x,-3Pi,3Pi}]
```



Remarque.- Un plus grand nombre de termes donne une image plus fidèle de f(x) = x périodique.

Exemple 5.1.8.

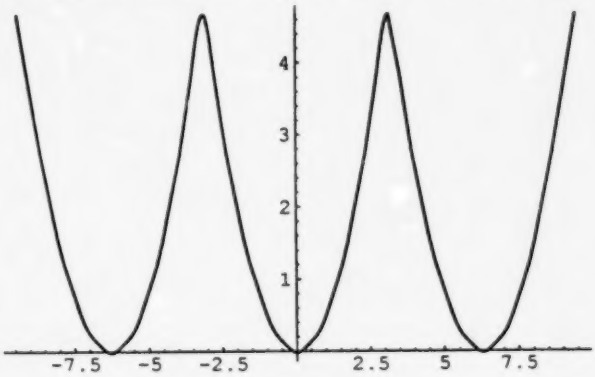
Développer en série de Fourier la fonction f(x) = $x^2/2$ (sept termes). Tracer le graphique.

```
Clear[fa,fb,j,n,f,x,s];
fa[j_]:=Integrate[f[x] Cos[j x],{x,-Pi,Pi}]/Pi;
fb[j_]:=Integrate[f[x] Sin[j x],{x,-Pi,Pi}]/Pi;
f[x_]:= x^2 /2;
s = Table[fa[j] Cos[j x]+ fb[j] Sin[j x],{j,1,7}];
```

```
AppendTo[s, fa[0]/2]
```

$$\{-2\,\mathrm{Cos}[x],\ \frac{\mathrm{Cos}[2\,x]}{2},\ \frac{-2\,\mathrm{Cos}[3\,x]}{9},\ \frac{\mathrm{Cos}[4\,x]}{8},\ \frac{-2\,\mathrm{Cos}[5\,x]}{25},\ \frac{\mathrm{Cos}[6\,x]}{18},\ \frac{-2\,\mathrm{Cos}[7\,x]}{49},\ \frac{\mathrm{Pi}^2}{6}\}$$

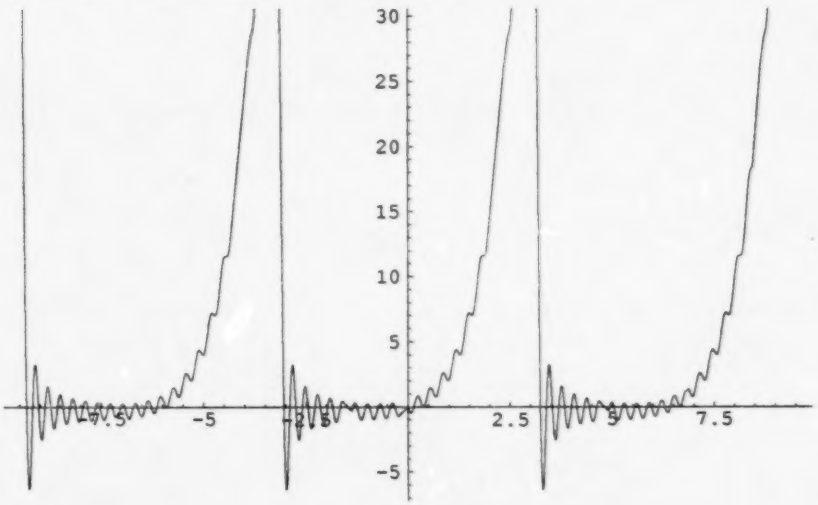
```
seriefourier[x_]:= Plus @@ s

Plot[seriefourier[x],{x,-3Pi,3Pi}]
```



Exemple 5.1.9.

Le développement de $f(x) = x\,e^x$ en série de Fourier.

```
Clear[fa,fb,j,n,f,x,s];
fa[j_]:=Integrate[f[x] Cos[j x],{x,-Pi,Pi}]/Pi;
fb[j_]:=Integrate[f[x] Sin[j x],{x,-Pi,Pi}]/Pi;
f[x_]:= x Exp[x];
s =Table[fa[j] Cos[j x]+ fb[j] Sin[j x],{j,1,20}];
AppendTo[s,fa[0]/2];
seriefourier[x_]:= Plus @@ s
Plot[seriefourier[x],{x,-3Pi,3Pi}]
```




```
-Graphics-
```

Remarques.- Toutes ces formules peuvent être généralisées pour donner les séries de Fourier de fonctions périodiques quelconques.

Une des remarquables propriétés des séries de Fourier est leur intégrabilité terme à terme. Nous avons le théorème suivant:

Si f(x) est continue par section et si f(x) est décomposée en série de Fourier alors cette série peut être intégrée terme à terme entre les deux bornes ξ et x appartenant à l'intervalle $-\pi \leq x \leq \pi$.
De plus, pour toute valeur fixe ξ la série converge uniformement en x.

En terminant ce chapitre, nous ne pouvons passer sous silence le fait que Wolfram Research donne la façon de trouver les fonctions les plus caractéristiques de *Mathematica* dans le livre *Guide to Standard Mathematica Packages*.

```
<<Calculus`FourierTransform`
```

```
?FourierExpSeries
```

```
FourierExpSeries[expression, {x, x0, x1}, n] donne le développement
d'ordre n en série exponentielle de expression comme une foncion
périodique sur l'intervalle de x0 à x1.
```

```
?FourierTrigSeries
```

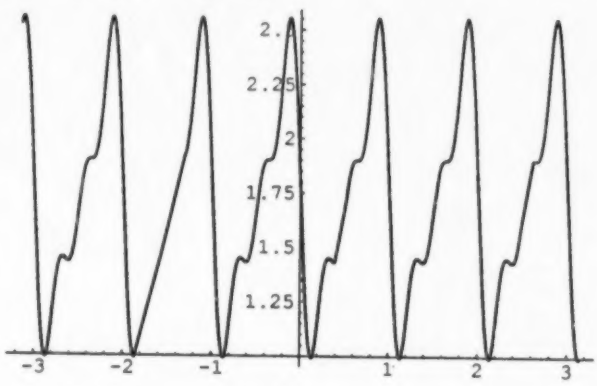
```
FourierTrigSeries[expression,{x, x0, x1}, n] donne le développement
d'ordre n en série trigonométrique de expression comme une fonction
périodique sur l'intervalle de x0 à x1.
```

Voici un exemple qui fait appel aux séries de Fourier avec la fonction $y = f(x) = e^x$.

```
FourierTrigSeries[ Exp[x],{x,0,1},3]
```

$$
\begin{aligned}
&-1 + E + 2\left(-\left(\frac{1}{1 + 4\,Pi^2}\right) + \frac{E}{1 + 4\,Pi^2}\right)\text{Cos}[2\,Pi\,x] + \\
&\quad 2\left(-\left(\frac{1}{1 + 16\,Pi^2}\right) + \frac{E}{1 + 16\,Pi^2}\right)\text{Cos}[4\,Pi\,x] + \\
&\quad 2\left(-\left(\frac{1}{1 + 36\,Pi^2}\right) + \frac{E}{1 + 36\,Pi^2}\right)\text{Cos}[6\,Pi\,x] + \\
&\quad 2\left(\frac{2\,Pi}{1 + 4\,Pi^2} - \frac{2\,E\,Pi}{1 + 4\,Pi^2}\right)\text{Sin}[2\,Pi\,x] + \\
&\quad 2\left(\frac{4\,Pi}{1 + 16\,Pi^2} - \frac{4\,E\,Pi}{1 + 16\,Pi^2}\right)\text{Sin}[4\,Pi\,x] + \\
&\quad 2\left(\frac{6\,Pi}{1 + 36\,Pi^2} - \frac{6\,E\,Pi}{1 + 36\,Pi^2}\right)\text{Sin}[6\,Pi\,x]
\end{aligned}
$$

```
Plot[%,{x,-Pi,Pi}]
```




```
-Graphics-
```

BIBLIOGRAPHIE

Cette bibliographie est à l'intention des lecteurs désireux d'approfondir le sujet.

Adams, Robert A, *Calcul Différentiel et Intégral dans l'espace*, Éditions Addison-Wesley, 1989.

Barnsley, Michael F; Devaney, Robert L; Mandelbrot, Benoit B; Peitgen, Heinz-Otto; Saupe, Dietmar; Voss, Richard F, *The Science of Fractal Images*, Springer-Verlag, 1988.

Barnsley, Michael, *Fractals Everywhere*, Academic Press Inc, 1988.

Bergamini, David, *Les Mathématiques*, traduit de l'anglais par Serge Ouvaroff. Collection Time-Life, 1965.

Casti, John L., *Alternate Realities*, JohnWiley & Sons, 1989.

Courant R., *Differential and Integral Calculus*, Interscience Publishers Inc, 1961.

Ghyka, Matila, *The Geometry of Art and Life*, Dover Publications, 1977.

Gleick, James, *Chaos*, Penguin Books, 1987.

Gray, Theodore W; Glynn, Jerry, *Exploring Mathematics with Mathematica*, Addison-Wesley Publishing Company Inc, 1991.

Haggerty, Gerald B., *Elementary Numerical Analysis with Programming*, Allyn and Bacon Inc, 1972.

Lauwerier, Hans, *Fractals: Meetkundige figuren in eindeloze herhaling*, traduit du hollandais par Sophia Gill-Hoffstädt, Princeton University Press, 1991.

Le Beux, Pierre, *Introduction au PASCAL*, Sybex, 1980.

Maeder, Roman, *Programming in Mathematica*, Addison-Wesley Publishing Company Inc, 1990.

Reumont, André; Reumont, Pierre, *Projet Mathétactiques*, Service de Développement Pédagogique du Collège de Maisonneuve, 1991.

Shenk, Al, *Calculus and Analytic Geometry*, Goodyear Publishing Company, 1977.

Tricot, Claude; Picard, Jean-Marc, *Ensembles et Statistique*, McGraw-Hill, 1969.

Wolfram, Stephen, *Mathematica*, première et deuxième éditions, Addison-Wesley Publishing Company Inc, 1991.

Index

A

Accroissement ... 102
Ackermann ... 88-90
Aires ... 84
Amplitude ... 224
AppendTo ... 35
ArcCos ... 31
Archimède ... 220
ArcSin ... 125

B

Bernouilli ... 213
Block ... 33-43
Blocs de décision ... 47
Boucles ... 34
Break ... 51

C

Calculateur ... 1
Centre de gravité ... 196
Chiffres romains ... 58-62-91
ComplexInfinity ... 174
Conditionnelles ... 47
Conoïde ... 208
Constantes ... 8
Continue ... 51
Continuité ... 109
ContourPlot ... 177-185
Cosinus directeurs ... 31-33
Croissance ... 100

D

D ... 14
Dali Salvador ... 82
Dames ... 73-77
Dérivée ... 14-118-119
Dérivée direct ... 178-179
Dérivée partielle ... 178
Dérivée totale ... 15-178
Différentielles ... 142
Do ... 35-40
Domaine ... 99

E

e ... 8-9-106
Eliminate ... 22
Erreur relative ... 143
Expand ... 12-69
Équation différentielle ... 212
Équation ... 18-19
Étiquettes ... 13

F

Facteur premier ... 7-98
Factor ... 13
FactorInteger ... 98
Ferma ... 135
Fibonacci ... 41-78
FindMinimum ... 23
FindRoot ... 23
FixedPoint ... 36-56
Flatten ... 90
Fonction abrégée ... 5
Fonction contrainte ... 186
Fonction factorielle ... 6
Fonction modulo ... 7
Fonction périodique ... 223
Fonction pure ... 68-70
Fonctionnelle ... 68
Fonctions trigonométriques ... 6
For ... 37
Fractale ... 42
Fréquence ... 213
Friction fluide ... 219
Frottement ... 216

G

Giration ... 197
Goto ... 51-53
Gradient ... 180-183-189
GraphicsArray ... 46-119
Graphiques ... 24

H

Hyperbolique ... 171
Hypercubes ... 79-80-94-95

Simplify ... 20-73
Simpson ... 154
Solve ... 19-20-189
Sommation ... 83
Somme ... 4-17
Spirale ... 77-78
StringJoin ... 59
Sum ... 15-17
Switch ... 47-49

T

Table ... 38
Tangente ... 118
ToExpression ... 59
Tours de Hanoï ... 93
Trapèzes ... 153
Triangle de Pascal ... 57
TrueQ ... 47

V

Variables ... 10
Vibration ... 224
Vitesse ... 123
Volumes ... 198

W

Which ... 47
While ... 36-56
Wolfram Stephen ... 2

Index

A

Accroissement ... 102
Ackermann ... 88-90
Aires ... 84
Amplitude ... 224
AppendTo ... 35
ArcCos ... 31
Archimède ... 220
ArcSin ... 125

B

Bernouilli ... 213
Block ... 33-43
Blocs de décision ... 47
Boucles ... 34
Break ... 51

C

Calculateur ... 1
Centre de gravité ... 196
Chiffres romains ... 58-62-91
ComplexInfinity ... 174
Conditionnelles ... 47
Conoïde ... 208
Constantes ... 8
Continue ... 51
Continuité ... 109
ContourPlot ... 177-185
Cosinus directeurs ... 31-33
Croissance ... 100

D

D ... 14
Dali Salvador ... 82
Dames ... 73-77
Dérivée ... 14-118-119
Dérivée direct ... 178-179
Dérivée partielle ... 178
Dérivée totale ... 15-178
Différentielles ... 142
Do ... 35-40
Domaine ... 99

E

e ... 8-9-106
Eliminate ... 22
Erreur relative ... 143
Expand ... 12-69
Équation différentielle ... 212
Équation ... 18-19
Étiquettes ... 13

F

Facteur premier ... 7-98
Factor ... 13
FactorInteger ... 98
Ferma ... 135
Fibonacci ... 41-78
FindMinimum ... 23
FindRoot ... 23
FixedPoint ... 36-56
Flatten ... 90
Fonction abrégée ... 5
Fonction contrainte ... 186
Fonction factorielle ... 6
Fonction modulo ... 7
Fonction périodique ... 223
Fonction pure ... 68-70
Fonctionnelle ... 68
Fonctions trigonométriques ... 6
For ... 37
Fractale ... 42
Fréquence ... 213
Friction fluide ... 219
Frottement ... 216

G

Giration ... 197
Goto ... 51-53
Gradient ... 180-183-189
GraphicsArray ... 46-119
Graphiques ... 24

H

Hyperbolique ... 171
Hypercubes ... 79-80-94-95

I

If ... 47-71-101-112
Indice ... 17
Infini ... 107
Inflexion ... 129
InputString ... 59
Integrate ... 14-16-86-148
Intégrales multiples ... 196-197

J

Jeu d'oeufs ... 63

L

Label ... 51
Langage ... 34
LCM ... 64
Length ... 50
Limit ... 19
Limite ... 18-105-107
ListPlot ... 98-99-107
Logarithme ... 6

M

Macintosh IIci ... 2
MacLaurin ... 160-162-167
Mandelbrot, Benoît ... 42
MatrixForm ... 58
Max ... 98
Max/Min ... 103
Maximum ... 186
Min ... 104
Minimum ... 186
Monte-Carlo ... 65

N

Nest ... 36
Newton-Raphson ... 54-56
NIntegrate ... 16-23
Nombre d'Or ... 42
Nombre premier ... 98
NRoots ... 23
NSum ... 23

O

Opérateurs ... 14

P

π ... 167
ParametricPlot ... 26-139-141
ParametricPlot3D ... 202
Paramétrique ... 135
Parenthèses ... 4-7
Partition ... 88
Phase ... 224
Plot ... 22
PlotGradientField ... 184
PlotRange ... 25
PlotStyle ... 106
Point d'intérogation ... 1
Polynôme ... 84
PrependTo ... 35
Précision ... 3
Primitives ... 146
Print ... 32-34-35-36-38
Procédures ... 30-47
Product ... 15
Programmation ... 53
  fonctionnelle ... 68
  mathématique ... 87
  procédurale ... 53
  récursive ... 87
Puissance ... 4-83

R

Racine ... 55
Racine carrée ... 6-55
Racine cubique ... 56
Racines ... 71-73
Random ... 43-65
Return ... 39-51
Résistance fluide ... 219
Résultante ... 217
Résultats antérieurs ... 10
Riemann ... 151
Roots ... 20

S

SameQ ... 47
Séries ... 156
Séries de Fourier ... 223

Simplify ... 20-73
Simpson ... 154
Solve ... 19-20-189
Sommation ... 83
Somme ... 4-17
Spirale ... 77-78
StringJoin ... 59
Sum ... 15-17
Switch ... 47-49

T

Table ... 38
Tangente ... 118
ToExpression ... 59
Tours de Hanoï ... 93
Trapèzes ... 153
Triangle de Pascal ... 57
TrueQ ... 47

V

Variables ... 10
Vibration ... 224
Vitesse ... 123
Volumes ... 198

W

Which ... 47
While ... 36-56
Wolfram Stephen ... 2